

Semantics, Language and Geometry: Learning to Understand the Scene

Iro Laina

Dissertation





Technische Universität München
Fakultät für Informatik
Lehrstuhl für Informatikanwendungen in der Medizin

Semantics, Language and Geometry: Learning to Understand the Scene

Iro Laina

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzende(r): Prof. Dr.-Ing. Darius Burschka

Prüfer der Dissertation:

1. Prof. Dr. Nassir Navab
2. Prof. Gustavo Carneiro, Ph.D.
3. Prof. Gregory D. Hager

Die Dissertation wurde am 29.04.2020 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 09.09.2020 angenommen.

Iro Laina

Semantics, Language and Geometry: Learning to Understand the Scene

Dissertation, Version 1.0.1

Technische Universität München

Fakultät für Informatik

Lehrstuhl für Informatikanwendungen in der Medizin

Boltzmannstraße 3

85748 and Garching bei München

Abstract

Thanks to their visual processing system, humans can understand complex visual content with remarkable speed and accuracy, grasping at just a single glance what is known as the gist of a scene. With the term *scene* we refer to a view of a complex real-world environment where various objects are arranged in meaningful ways and humans can act within. If we are to build intelligent systems, they should be equipped with the same fundamental capability to deeply understand their surroundings, which becomes the enabling factor for higher-level tasks such as navigating and acting in the environment. This makes scene understanding one of the primary goals of computer vision encompassing a multitude of perceptual and semantic tasks; from recovering the 3D world from a 2D view to recognizing and localizing objects, their attributes and their relationships.

In this dissertation, we address tasks related to the field of scene understanding by employing deep learning methods. First, we propose a fully convolutional residual architecture to learn depth estimation from a single image in indoor and outdoor environments. Subsequently, we extend this architecture to semantic image segmentation. The potential of learning structure and semantics is further demonstrated by building a hybrid system for simultaneous localization and mapping. The outcome is a dense, real-scale and semantically meaningful 3D reconstruction of the environment. We then move from scene-centric to object-centric localization addressing two practical scene understanding scenarios in robotics, detection and segmentation of articulated instruments and robotic grasp detection.

In the second part of the dissertation, we emphasize the importance of communication in intelligent agents and thus address scene understanding from a perspective that lies on the intersection of computer vision and natural language processing. One relevant application for scene understanding is generating image descriptions, which is of high practical impact, for example to assist visually impaired users. We propose an unsupervised way to tackle this problem which allows to exploit varying text corpora and image sources without relying on manually annotated paired data. In the core of our approach lies a multi-modal embedding space in which visual and linguistic representations co-exist indistinguishably.

Despite remarkable progress in scene understanding, agents deployed in the real-world are likely not going to be perfect at the task they have been designed for; for example, the agent might come across objects it does not recognize. With our last contribution we address this problem by enabling interaction between the user and the agent. We leverage human perception, and specifically hints given by the user in natural language, and build a mechanism to guide the network into re-evaluating its inference. We demonstrate this mechanism on the task of semantic segmentation, though it can also find application in other tasks as well as lifelong learning.

Zusammenfassung

Dank seines visuellen Verarbeitungssystems kann der Mensch komplexe visuelle Inhalte mit bemerkenswerter Geschwindigkeit und Genauigkeit wahrnehmen, verstehen und mit einem einzigen Blick das Wesentliche einer Szene erfassen. Mit Szene bezeichnen wir eine komplexe, reale Umgebung, in der verschiedene Objekte auf sinnvolle Weise angeordnet sind und in der Menschen handeln können. Wenn wir intelligente Systeme bauen wollen, sollten sie mit der gleichen Fähigkeit ihre Umgebung zu verstehen ausgestattet sein. Dies ist der entscheidende Faktor für übergeordnete Aufgaben wie das Navigieren und Handeln in der Umwelt. Damit wird das Verstehen von Szenen zu einem der primären Ziele des Computer Vision, das eine Vielzahl von Wahrnehmungs- und semantischen Aufgaben umfasst; von der Wiederherstellung der 3D-Welt aus einer 2D-Ansicht bis zum Erkennen und Lokalisieren von Objekten mit ihren Eigenschaften und Beziehungen.

In dieser Dissertation behandeln wir Herausforderungen aus dem Bereich des Szenenverständnisses durch Einsatz von Methoden des tiefen Lernens (Deep Learning). Zunächst führen wir eine Netzarchitektur ein, um Tiefenschätzung von einem einzelnen Bild für Innen- und Außenbereiche zu lernen. Anschließend erweitern wir diese Architektur für semantische Bildsegmentierung. Außerdem demonstrieren wir das Lernen von Struktur und Semantik indem wir ein hybrides System zur simultanen Positionsbestimmung und Kartenerstellung beschreiben. Das Ergebnis ist eine dichte und semantische 3D-Rekonstruktion der Umgebung. Anschließend gehen wir von der szenenzentrierten zur objektzentrierten Lokalisierung über und befassen uns mit zwei praktischen Szenarien in der Robotik, der Erkennung und Segmentierung von gelenkigen Instrumenten und der Roboter-Greifererkennung.

Wir analysieren Agenten, die in der Lage sind, zu kommunizieren. Diese Aufgabe liegt in der Schnittmenge von Computersehen und Sprachverarbeitung. Eine relevante Anwendung ist die Generierung von Bildbeschreibungen, um beispielsweise sehbehinderten Benutzern zu helfen. Wir schlagen eine unüberwachte Methode zur Lösung dieses Problems vor, die es ermöglicht, verschiedenste Textkorpora und Bilder zu nutzen ohne manuell annotierte, gepaarte Daten zu verwenden. Im Kern unseres Ansatzes liegt ein multimodaler Einbettungsraum, in dem visuelle und sprachliche Darstellungen ununterscheidbar nebeneinander existieren.

Trotz bemerkenswerter Fortschritte im Szenenverständnis, die in der realen Welt eingesetzt werden, höchstwahrscheinlich nicht perfekt für die Aufgabe, für die sie konzipiert wurden; so könnte der Agent beispielsweise auf Objekte stoßen, die er nicht kennt. Mit unserem letzten Beitrag adressieren wir dieses Problem, indem wir Interaktion zwischen Benutzer und Agenten ermöglichen. Wir nutzen die menschliche Wahrnehmung und sprachlichen Hinweise des Benutzers und leiten das Netzwerk dazu an, die Vorhersage neu zu bewerten.

Acknowledgments

The life of a PhD student is sometimes compared to a roller-coaster ride; which, in a way, does not lie far from the truth. Reaching the end of this ride, I am very thankful to everyone who encouraged me to jump in and supported me during this time in any way — morally or materially. The work carried out for this dissertation would not have been possible without great colleagues, friends and family.

To begin with, I would like to express my gratitude to my advisor, Prof. Nassir Navab, not only for giving me this chance in the first place, his support and advice, but also for the scientific freedom he trusted me with, which allowed me to pursue problems I was genuinely interested in. I would also like to thank Prof. Gregory Hager for his collaboration, guidance and the inspiring discussions I had with him during these years.

I would also like to thank my senior colleagues Maximilian Baust and Federico Tombari for their mentorship, support and encouragement, especially during hard times; and Vasilis Belagiannis for introducing me to the world of deep learning as a Master's student. And a thank you to Martina Hilla and Manuela Fischer for always taking care of all sorts of administrative things.

One thing that I am undoubtedly grateful for is getting to spend my PhD years with brilliant colleagues and that my colleagues have become my friends. Nicola Rieke, Marco Esposito, Jakob Weiss, Beatrice Demiray, Helisa Dhamo, Fabian Manhardt, David Tan and Keisuke Tateno thank you for your support and collaboration, for mutual hardships, for fun travels and times together. A special shout-out goes to Salvatore (Jack) Virga who shared over six years of this journey with me in Munich and Ari Tran who joined our journey a little later; thank you for always being there, for being great friends, for many unforgettable adventures and — it goes without saying — for all the pizza.

Finally, I am deeply thankful for my family and the support they have given me throughout my life, which has brought me where I am today. And to Christian Rupprecht, my partner in life and at work, goes a very special thank you. Your love, support, ideas and never-ending positivity and patience throughout every moment of this ride have made it all possible.

Contents

I INTRODUCTION

1	Introduction	3
1.1	Motivation	3
1.1.1	Perception	4
1.1.2	Communication	4
1.1.3	Interaction	6
1.2	Scene Understanding	7
1.3	Outline of Dissertation	8
2	Contributions	11
3	Theory and Fundamentals	13
3.1	Deep Learning	13
3.1.1	A Note on History	13
3.2	Neural Networks	14
3.2.1	Minimal Example	15
3.2.2	Convolutional Neural Networks	17
3.2.3	Recurrent Neural Networks	20
3.3	Supervised and Unsupervised Learning	21
3.4	Transfer Learning	22

II SCENE UNDERSTANDING THROUGH GEOMETRY AND SEMANTICS

4	Learning Geometry and Semantics	25
4.1	Introduction	25
4.1.1	Motivation	25
4.1.2	Contribution	26
4.2	Related Work	27
4.3	Depth Estimation from a Single Image	31
4.3.1	Fully Connected versus Fully Convolutional	31
4.3.2	Fully Convolutional Residual Network (FCRN)	32
4.3.3	Loss Function	36
4.4	Semantic Segmentation	38
4.4.1	Architecture Modification	39
4.5	Experimental Results	39

4.5.1	Experiments on Depth Estimation	41
4.5.2	Applications of Depth Estimation	47
4.5.3	Experiments on Semantic Segmentation	49
4.6	Conclusion	52
5	Learning in Scene Reconstruction	55
5.1	Introduction to SLAM	55
5.2	Related Work	57
5.3	Monocular Semantic SLAM	60
5.3.1	Camera Pose Estimation	61
5.3.2	Keyframe Processing	62
5.3.3	Depth Refinement	63
5.3.4	Semantic Reconstruction	64
5.4	Results and Evaluation	65
5.5	Conclusion	68
6	Object-centric Understanding	71
6.1	Motivation	71
6.2	Localization of Surgical Instruments	73
6.2.1	Methodology	73
6.2.2	Results and Evaluation	76
6.3	Localization of Grasping Points	81
6.3.1	Methodology	82
6.3.2	Results and Evaluation	86
6.4	Conclusion	92

III NATURAL LANGUAGE IN SCENE UNDERSTANDING

7	Image Captioning: Language as Output	95
7.1	Introduction	95
7.1.1	Motivation	95
7.1.2	Contribution	96
7.2	Related Work	98
7.3	Unsupervised Image Captioning	102
7.3.1	Language Model	103
7.3.2	Domain Alignment	106
7.4	Experiments and Results	111
7.4.1	Implementation Details	111
7.4.2	Unpaired Setting	113
7.4.3	Unsupervised Setting	118
7.4.4	Visualization of Embedding Space	119
7.5	Limitations and Discussion	121
7.6	Conclusion	123
8	User Interaction: Language as Input	125

8.1	Introduction	125
8.1.1	Motivation	126
8.1.2	Contribution	127
8.2	Related Work	128
8.3	Interaction as Feature Guiding	131
8.4	Guiding by Back-propagation	134
8.4.1	Method	134
8.4.2	Experiments	135
8.5	Guiding in Natural Language	136
8.5.1	Method	136
8.5.2	Experiments	140
8.5.3	Visualization of Guiding Vectors	145
8.6	Conclusion and Outlook	147

IV CONCLUSION

9	Conclusion	151
9.1	Summary of Findings	151
9.2	Future Outlook	153
	Authored and Co-authored Publications	155
	Bibliography	157
	List of Figures	189
	List of Tables	195

Part I

Introduction

Introduction

1.1	Motivation	3
1.1.1	Perception	4
1.1.2	Communication	4
1.1.3	Interaction	6
1.2	Scene Understanding	7
1.3	Outline of Dissertation	8

1.1 Motivation

Artificial intelligence (AI) is a discipline that has attracted immense interest in the scientific community since the birth of this term in 1956 (John McCarthy, Dartmouth Conference), though the practical definition and goals of AI have evolved throughout the years. In the early stages of AI, the problems that were addressed were easy to solve by computers, although challenging for humans; these were formal problems that could be described by a set of rules, such as proving theorems (Geometry Theorem Prover, H. Gelernter, 1959) or playing checkers (A. Samuel, 1952) [361]. As more and more tasks were successfully solved by machines, the type of problems that were regarded as requiring intelligence and the definition of intelligence itself changed. From then till the present day, we face a paradigm shift, as AI research is now concerned with problems that are intuitive and natural to humans, but difficult to address by machines; for example, effortlessly recognizing other people, objects and structures and being able to reason.

Without a doubt, we are still very far—if even achievable—from human-level or general AI, *i.e.* machines that are capable of sensing, understanding and learning tasks just as humans are able to; machines that can accomplish general tasks outside a narrow field of application. Although to achieve this goal as a whole is challenging to say the least, it is now possible to address intuitive problems or sub-problems, such as the ones mentioned above, with reasonable success. The solution to such problems has its roots in machine learning, *i.e.* we let machines *learn* abstract concepts from experience by providing them with several training examples [128]. This brings us one step closer to intelligent agents which can act autonomously, *i.e.* without relying solely on prior knowledge specified by the designer.

Regardless of the level of autonomy or which tasks are deemed to require “intelligence” at a given point in time, an agent should have to ability to perceive its environment (through sensors) and act upon it (through actuators). In this dissertation, we define and discuss a set of fundamental capabilities that can be seen as a stepping stone for intelligent agents.

1.1.1 Perception

“ perception, *noun*

1. the way you notice things, especially with the senses
2. the ability to understand the true nature of something
3. an idea, a belief or an image you have as a result of how you see or understand something

— *Oxford Advanced Learner's Dictionary* ”

Perception is the ability to understand or become aware of the environment using the senses, or some sensory input when it comes to agents. In the context of this dissertation, we discuss visual perception, although other kinds also exist (*e.g.* auditory or haptic).

In human beings, visual perception is the ability to interpret their surroundings through vision. In biological vision, an image is formed on the retina as light reflected from the surrounding world reaches the eye. Interpreting the visual input implies some form of conscious or unconscious *understanding*. According to Hermann von Helmholtz and his theory on “unconscious inference” (1867), perception is not only a product of sensory information on the retina but also of our learned experience about the world.

Making machines capable of perception has been the core objective in the field of computer vision. Initially, in the early 1970s, visual perception was considered an easy first step towards the more complicated AI problems. Marvin Minsky considered it to be doable as a summer project for the undergraduate student Gerald Jay Sussman: “spend the summer linking a camera to a computer and getting the computer to describe what it saw” [42]. Now, 50 years later, image understanding and visual perception is still not considered a solved problem. We have, however made significant progress.

1.1.2 Communication

“ communication, *noun*

1. the activity or process of expressing ideas and feelings or of giving people information
2. methods of sending information, especially telephones, radio, computers, etc. or roads and railways
3. a message, letter or telephone call

— *Oxford Advanced Learner's Dictionary* ”

Communication can be achieved through various means, from signs to electromagnetic signals; *verbal* communication, in particular, refers to conveying information through *language*, spoken, written or gestured. If intelligent agents are to be used in collaboration with or to assist humans, they will be often required to communicate their understanding about the world, provide information or act as a conversational partner (*e.g.* chatbots). Further, they

should be able to do so in a way that is interpretable by people, for example in a human language.

Michael Halliday (1975) identifies seven functions of language in the early years of childhood [146]:

- **Instrumental:** to express and satisfy one's needs
- **Regulatory:** to control the behavior of others or tell them what to do
- **Interactional:** to facilitate interaction with others and form relationships
- **Personal:** to express one's personal opinions, feelings or identity
- **Imaginative:** to create imaginary content, such as telling stories
- **Heuristic:** to discover, to acquire knowledge about the world
- **Representational:** to communicate information to others

Halliday believes that, in children, language develops in order to satisfy motives related to these functions. We argue that language can serve similar functionalities in intelligent agents. In fact, an agent with language capabilities is fundamental for various reasons. We can thus generalize the above definitions to machines capable of language as follows.

Representational: The use of language to convey information is central to intelligent systems. As an example, we can think of automatic generation of image descriptions to aid visually impaired users [189, 228], conversational agents (chatbots) and voice-enabled home assistants.

Heuristic: The use of language to acquire knowledge. One way to autonomy is an ever-growing self-acquired knowledge base. Language is a critical skill that equips agents with the potential of lifelong learning and continuously expanding their knowledge as they explore their environment, for example by asking questions about the world [294, 373, 424, 462], or parsing the web (*e.g.* Wikipedia).

Regulatory: Regulatory language has been used extensively even in early computers. The user needs to know how to make the system work, what data it needs and which inputs it requires; the more descriptive and accurate the language is (*e.g.* hints or error messages), the better.

Imaginative: There already exist instances of artificially generated novels, poems, jokes or textbooks¹, which directly relate to the imaginative function. Due to the challenges of language modeling, these are not yet strong enough to pass the Turing test. Recently, however, significant progress has been made in language generation using extremely large language models [48].

Interactional: This is the part of language that is currently mostly used in games. AI-controlled characters that express themselves realistically towards the player create a greater emotional connection between the user and the game. This naturally extends to other virtual characters that we interact with. However, we are still only at the beginning of developing systems that can naturally interact with humans.

¹Besides language, there are also instances of computer-generated art, music, etc.

Personal: Without a doubt, computers do not have a “personality” in the way humans do. However, the effort towards safe and *explainable* AI (XAI) requires systems that are interpretable by design [122], perhaps systems that can generate an explanation of their own behavior and decision making process in a human-readable format, *e.g.* natural language. The ability of agents to express themselves can contribute to gradually building trust in such systems, especially when it comes to safety-critical applications.

1.1.3 Interaction

“ **interaction**, *noun*

1. the act of communicating with somebody, especially while you work, play or spend time with them
2. if one thing has an interaction with another, or if there is an interaction between two things, the two things have an effect on each other

— *Oxford Advanced Learner’s Dictionary* ”

The ability to interact is of paramount important to intelligence. Interactive behavior might involve acting and/or receiving feedback from the environment (or objects that lie within), humans or other agents. There exist several manifestations of interaction in learning and AI. This is, for example, the foundation of the fields of human-robot collaboration [30], reinforcement and imitation learning. In reinforcement learning [391], the agent interacts with its environment to gather experience, *i.e.* taking actions that optimize a reward function. On the other hand, the goal of imitation learning [177] is to teach an agent complex tasks through demonstration.

These examples are outside the scope of this thesis and will not be discussed in detail. In fact, it seems very hard to tackle such higher-level reasoning tasks without a solid scene understanding system—which is in fact the focus of this dissertation. Thus, we hereby address interaction from perspectives that are more related to visual perception and communication.

From this perspective, when deployed into the real world, an agent will most likely not be perfect for the task it is designed for, inevitably making mistakes, *e.g.* due to a distributional shift, thus creating the problem of adaptation to the specifics of a new environment. For instance, it might not be able to recognize some objects. Here, an interactive system is the ideal bridge between the user and the algorithm. The goal is to leverage the superior capabilities and experience of the human in order to overcome the agent’s limitations. Enabling an interactive system means enabling the agent to identify its shortcomings (*e.g.* through uncertainty), being able to communicate them to the user and finally process their feedback for self-improvement.

1.2 Scene Understanding

In this dissertation, we will approach some aspects of the aforementioned principles and provide solutions in the context of scene understanding. Scene understanding has been a long-standing problem in computer vision and is so broad that it cannot be described with a universal definition. Generally speaking, scene understanding is involved with analyzing semantic and structural properties of a scene. Hereby we use the term *scene* to refer to a view of a complex real-world environment in which various objects are arranged in meaningful ways and entities (humans or agents) act within [450].

Human perception studies have shown that the visual system can process complex visual content with remarkable speed and accuracy [105, 324, 325, 369, 406]. In pioneering work conducted by Mary Potter [324, 325], presenting observers with a very quick succession of visual content, experiments showed that it takes under 200ms to understand the type of a scene and a lot of related visual information — such as the global structure and layout or sometimes major objects existing in the scene. The visual information that is perceived with just a quick glance is called the *gist* of the scene [310, 324]. The gist carries both perceptual and conceptual (semantic) information. The perceptual gist is related to structural information which is comprehended almost instantly during perception, while the conceptual gist is related to semantic information that is understood during or after viewing a scene [310].

The distinction between perceptual and conceptual understanding is also seen in neuroscience. The widely accepted two-streams hypothesis [127] describes the human vision system as the combination of two pathways that begin from the occipital lobe of the cerebral cortex. The “what pathway” (ventral stream) is responsible for object identification and recognition, while the “where pathway” (dorsal stream) processes the object’s spatial location relative to the viewer.

From a computational perspective, general and scalable scene understanding is considered very challenging. For this reason, we can usually identify multiple levels of “understanding”. These range from low-level perceptual tasks, such as detection of edges or intrinsic image properties, to higher-level semantic tasks to tasks that require reasoning “behind” the scene. The field of scene understanding is thus divided further into several dominant sub-fields that deal with classifying the scene and its attributes (at a coarse level); or estimating the structure or layout of the environment, recognizing and localizing objects and understanding their affordances, physical properties or their in-between interactions (at a finer level). Often part of the literature on total scene understanding addresses the problems related to the spatial or 3D component of the scene [139, 162], while another part focuses on visual recognition tasks [241]. Finally, unified scene parsing focuses on recognizing multiple visual concepts from various “levels” in parallel, *i.e.* scene types, objects and their parts, and also their material properties and textures [451].

It is important to note that truly understanding the scene goes far beyond recognition. The human visual system does not just perceive visual information but has the ability to *reason* about it. For example, using common sense, we can effortlessly infer other people’s actions or how these affect or are affected by their surroundings, reasoning about unobserved

causes or future events. Common-sense reasoning is what makes the difference between a machine that can only infer structural and semantic information and one that can answer visual questions about the scene, such as “*who is winning this football match?*” or “*will I need an umbrella today?*”. It is clear that to answer these questions the system would require the ability to combine various cues and possess higher-level cognitive skills and situational awareness.

Finally, it is worth noting that humans can rely on their abilities for compositionality and transferability to solve higher-level reasoning problems. We divide complex problems into individual components, some of which we have already learned for other tasks. Another cornerstone of human reasoning comes in fact from other modalities, such as language, audio or haptics, which are combined with and contribute to the learned visual knowledge and understanding. While such high-level reasoning is not yet in reach for machine learning systems, this dissertation provides some building blocks towards machines that can holistically understand the scene and communicate — about the scene — with humans.

1.3 Outline of Dissertation

The remaining chapters of this dissertation are organized and summarized as follows:

- Chapter 2** A list of publications on which this dissertation is based.
- Chapter 3** Deep learning is a major component of all methods presented in this dissertation. Thus, before we present our algorithms in detail, we give a brief introduction of the fundamental principles of deep learning and neural networks that are used throughout this work.
- Chapter 4** We first address image understanding by estimating the geometry and semantics of a scene from visual data (single images). We introduce a deep fully convolutional network for image-to-image translation problems, which we evaluate on the tasks of depth estimation and semantic segmentation.
- Chapter 5** Our next observation is that the *learned* geometry and semantic segments of the scene provide complementary information to *geometric* approaches for simultaneous localization and mapping. We propose an approach to reconstruct 3D environments from monocular videos by integrating these two knowledge sources.
- Chapter 6** In this chapter, we move from complex views down to scenes focused on individual objects, thus discussing object-level understanding. We present two problems where this is applicable: localization of instruments (*e.g.* surgical, robotic) and localization of grasping points around objects (*e.g.* as part of the visual perception system of a robotic arm.)
- Chapter 7** Natural language is an essential skill for communication between users and machines. Communicating about visual content, however, is a task

that requires recognition and generation across two modalities. To relax the need for aligned vision-and-language datasets, in this chapter, we tackle unsupervised image captioning, without image-caption pairs.

Chapter 8

Another important skill for intelligent systems is interaction with humans, for example leveraging user's feedback for self-improvement. Here, we revisit the task of semantic segmentation and describe a mechanism to retrospectively improve the system's performance, given hints from the user in natural language or other forms.

Chapter 9

We conclude with a summary of the contributed research, discussion of the limitations of current systems and future outlook.

Contributions

This dissertation is divided into two major thematic parts. In Part II we focus on visual perception problems related to scene understanding, such as depth estimation and semantic segmentation, which we address with novel deep learning techniques. This part builds on the following authored and co-authored publications:

- [Deeper Depth Prediction with Fully Convolutional Residual Networks](#)
Iro Laina*, Christian Rupprecht*, Vasileios Belagiannis, Federico Tombari, Nassir Navab.
In International Conference on 3D Vision (3DV), 2016.
- [CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction](#)
Keisuke Tateno*, Federico Tombari*, **Iro Laina**, Nassir Navab.
In Computer Vision and Pattern Recognition (CVPR), 2017.
- [Concurrent segmentation and localization for tracking of surgical instruments](#)
Iro Laina*, Nicola Rieke*, Christian Rupprecht, Josue Page Vizcaino, Abouzar Eslami, Federico Tombari, Nassir Navab.
In International Conference on Medical Image Computing and Computer-Assisted Interventions (MICCAI), 2017.
- [Dealing with Ambiguity in Robotic Grasping via Multiple Predictions](#)
Ghazal Ghazaei, **Iro Laina**, Christian Rupprecht, Federico Tombari, Nassir Navab, Kianoush Nazarpour.
In Asian Conference on Computer Vision (ACCV), 2018.

The research listed above finds application in systems that require holistic visual perception, such autonomous vehicles, home or industrial robotics, computer-aided surgery, etc.

Moving to Part III, our reasoning is that for intelligent visual systems to be used in collaboration with or to assist humans, they must be often able to express and communicate their understanding in a natural way. Thus, the second part of this dissertation focuses on problems that lie at the intersection of computer vision and natural language and builds on the following publications:

- [Towards Unsupervised Image Captioning with Shared Multimodal Embeddings](#)
Iro Laina, Christian Rupprecht, Nassir Navab.
In International Conference on Computer Vision (ICCV), 2019.

* denotes equal contribution.

- [Guide Me: Interacting with Deep Networks](#)
Christian Rupprecht*, **Iro Laina***, Nassir Navab, Gregory D. Hager, Federico Tombari.
In Computer Vision and Pattern Recognition (CVPR), 2018.

Finally, the following co-authored publications will not be extensively discussed in this dissertation:

- [Learning in an Uncertain World: Representing Ambiguity Through Multiple Hypotheses](#)
Christian Rupprecht, **Iro Laina**, Robert DiPietro, Maximilian Baust, Federico Tombari, Gregory D. Hager, Nassir Navab.
In International Conference on Computer Vision (ICCV), 2017.
- [Peeking behind objects: Layered depth prediction from a single image](#)
Helisa Dharmo, **Iro Laina**, Keisuke Tateno, Nassir Navab, Federico Tombari.
In Pattern Recognition Letters 125, 2019.
- [Semantic Image Manipulation Using Scene Graphs](#)
Helisa Dharmo*, Azade Farshad*, **Iro Laina**, Nassir Navab, Gregory D. Hager, Federico Tombari, Christian Rupprecht.
In Computer Vision and Pattern Recognition (CVPR), 2020.

Theory and Fundamentals

3.1	Deep Learning	13
3.1.1	A Note on History	13
3.2	Neural Networks	14
3.2.1	Minimal Example	15
3.2.2	Convolutional Neural Networks	17
3.2.3	Recurrent Neural Networks	20
3.3	Supervised and Unsupervised Learning	21
3.4	Transfer Learning	22

3.1 Deep Learning

As mentioned in the previous chapter, we are currently able to design systems that alleviate the need for prior knowledge provided by the designer, such as hand-crafted features and rule-based programs. In fact, a truly *autonomous* system must be able to (continuously) acquire its own knowledge about the world instead of being dependent on human-provided knowledge. The field of **machine learning** deals with algorithmic or statistical solutions to problems, which a computer system learns from representations of data (*e.g.* images or features). Until recently, many tasks were addressed by designing and providing hand-crafted features into simple machine learning algorithms. Instead, we can also let the computer learn abstract concepts (features) in a hierarchical fashion, *i.e.* progressively discovering more complex concepts (higher abstraction) by building on simpler ones, thus building *deep* models which consist of many layers of abstraction; this gave rise to the now widely used term **deep learning** [128].

In this chapter, we briefly introduce deep learning and neural networks as they are an integral part of the methods presented later in the dissertation. We focus on the building blocks that will be used in our methods. For a detailed mathematical introduction to deep learning we refer the reader to the book of Goodfellow et al. [128].

3.1.1 A Note on History

Deep learning is a term that was only recently popularized, but in fact it has existed under different umbrella terms and perspectives since the 1940s. There have been three areas/waves in AI and deep learning.

The first wave started with theoretical models of the brain, when McCulloch and Pitts [287] introduced the artificial **neuron** which was modeled as a linear function. In 1958, Rosenblatt [348] introduced the **perceptron**, which was the first practical implementation that enabled training of a single neuron through potentiometers. However, it soon became clear linear models have limitations; for example, as it was shown that in 1969, the perceptron cannot learn the XOR function [293] — and with that came the first winter in AI as the great visions of AI scientists could not be materialized.

The renaissance began in the 1980s with a movement known as connectionism, *i.e.* models of inter-connected computational units (neural networks). From a biological perspective, this corresponds to the nervous system, and from a computational perspective to hidden units. A major contribution to the resurgence of AI was the use of the generalized delta rule (**back-propagation**) [352] for training feed-forward neural networks — although not plausible learning rule from a biological perspective [285]. Back-propagation was an efficient algorithm to adjust the weights of a neural network by back-propagating errors and remains to date a core principle in deep learning. However, another AI winter followed as the ambitions and promises turned out to be more complicated to carry out as initially thought.

The third wave of AI, which we are still experiencing now in the year 2020, began with deep belief networks by Hinton et al. [157] (2006). Powered by the increased availability of data and as well as computational resources, researchers were now able to study and prove the importance of *depth* in neural networks [78]. Finally, the big revolution reached computer vision in 2012, when AlexNet [215], a deep architecture trained on GPUs, won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), surpassing competing methods by a significant margin (reducing the top-5 error from 26.1% down to 15.3%).

3.2 Neural Networks

Artificial neural networks (ANNs) are inspired by the biological brain. The main building component of ANNs is the neuron, which acts as a logical threshold gate of a linear function of the received input ($\mathbf{x} \in \mathbb{R}^N$, weights $\mathbf{W} \in \mathbb{R}^{N \times M}$ and (optionally) a bias $\mathbf{b} \in \mathbb{R}^M$).

$$f(\mathbf{W}, \mathbf{x}) = W_1x_1 + W_2x_2 + \dots + W_Nx_N + \mathbf{b} = \mathbf{W}^T \mathbf{x} + \mathbf{b} \quad (3.1)$$

The weights and biases of neurons are learnable, *i.e.* they can be adjusted as part of a training process. Neurons connect to each other and form a network inspired by biological neural networks. In **feed-forward** neural networks, the information flows from the input towards the output (forward pass), without directed loops or cycles. Neural network units are typically structured into groups (*layers*). One example is the **multi-layer perceptron** (MLP), which consists of an input and an output layer and one or more *hidden* layers.

However, if we use linear functions for neurons, the mapping of input to output through a multi-layer network is equivalent to a single-layer network, because a composition of linear functions is also a linear function. Thus, it is important to use *non-linear activation functions*

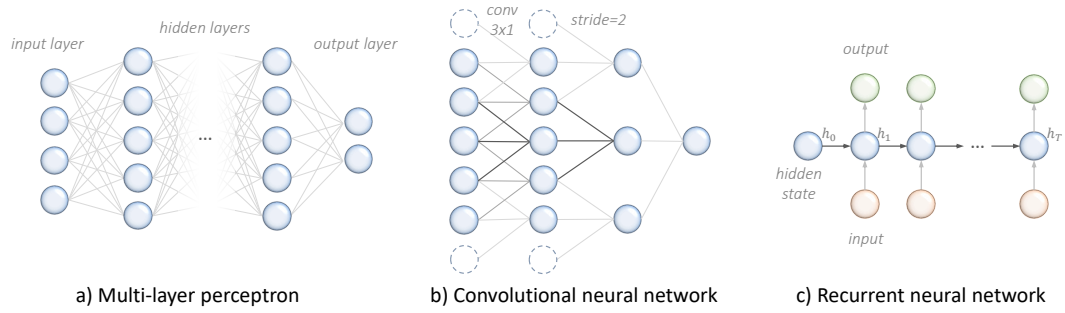


Figure 3.1 Neural network architectures. **a)** An MLP as a fully-connected feed-forward neural network. **b)** A CNN with 3×1 convolutions with zero padding and stride 1 and 2 respectively. A neuron in the second layer has a receptive field of 5 and thus “sees” the entire input vector. The last layer is fully-connected. **c)** A simple RNN; the hidden state h_t depends on the state at time step $t - 1$.

$\phi(\cdot)$ in between the layers. ϕ can for example be the sigmoid function, hyperbolic tangent or, most commonly in convolutional neural networks (Section 3.2.2), the Rectified Linear Unit (ReLU) [125, 304]—although other variations also exist [69, 152, 270].

3.2.1 Minimal Example

We will now build a simple two-layer example of a neural network to explain some fundamental concepts before moving to deep networks in the following sections.

Consider a function $f(\cdot)$ that maps an input $\mathbf{x} \in \mathbb{R}^{100}$ to an output $\mathbf{y} \in \mathbb{R}^2$. For example, \mathbf{x} could be some weather statistics and \mathbf{y} the estimated temperature at day and night for this day. With this example we will introduce the basics of machine learning.

We are given a dataset (e.g. weather observations) consisting of N samples $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ where $\mathbf{x}^{(i)} \in \mathbb{R}^{d_x}$, $\mathbf{y}^{(i)} \in \mathbb{R}^{d_y}$ and $i \in \{1, \dots, N\}$; this is a case of *supervised* learning. The learning process consists in finding a mapping $f_\theta(\mathbf{x}) = \hat{\mathbf{y}}$ with minimal error by observing the training examples $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$. With θ we denote the learnable parameters of f . We can see this process as finding an optimal set of parameters θ^* as measured by a criterion or **cost function**¹ \mathcal{L} on the training set:

$$\theta^* = \operatorname{argmin}_{\theta} \sum_{i=1}^N \mathcal{L} \left(f(\mathbf{x}^{(i)}), \mathbf{y}^{(i)} \right). \quad (3.2)$$

In this example, since day and night temperature are continuous values, we are dealing with a regression problem. A standard cost function for these types of problems is the \mathcal{L}_2 norm:

$$\mathcal{L}_2 \left(f(\mathbf{x}^{(i)}), \mathbf{y}^{(i)} \right) = \|f(\mathbf{x}^{(i)}) - \mathbf{y}^{(i)}\|_2^2, \quad (3.3)$$

¹we will also refer to this as a loss function throughout the dissertation

i.e. minimizing the squared Euclidean distance between the predicted vector $f(\mathbf{x}^{(i)})$ and the expected outcome $\mathbf{y}^{(i)}$ for sample i .

However, the training error does not provide sufficient information about how well the model will perform on previously *unseen* samples when deployed in the real world. During training, it is common to withhold a subset of data (*validation set*) which are not used to optimize the network, but to measure the network's generalization performance on unseen data. The term **regularization** refers to techniques that aim at improving the network's generalization performance, for example the \mathcal{L}_2 -norm of the parameters can be used as a regularizer $R(\theta) = \|\theta\|_2^2$ in addition to the cost function.

Other approaches include dropout (Section 3.2.2) or augmenting the data with various task-specific transformations.

Optimization

Let us now define $f(\cdot)$ as a two-layer neural network to solve the weather prediction problem.

$$f_{\theta}(\mathbf{x}) = \mathbf{W}_2 \phi(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2, \quad (3.4)$$

with parameters $\theta = (\mathbf{W}_1, \mathbf{b}_1, \mathbf{W}_2, \mathbf{b}_2)$, where \mathbf{W}_1 and \mathbf{W}_2 are matrices and \mathbf{b}_1 and \mathbf{b}_2 are vectors. As mentioned before, ϕ is a non-linear function. Here we have dropped the notation of the i -th sample ($\mathbf{x}^{(i)}$) for simplicity.

To find the optimal θ^* , we need to update all parameters from their initial values given the training samples. This is a difficult problem but gradient-based optimization can find a locally optimal solution through **gradient descent** [53]:

$$\theta^{(t+1)} = \theta^{(t)} + \lambda \frac{\partial \mathcal{L}_{\text{total}}}{\partial \theta^{(t)}} \quad (3.5)$$

and

$$\frac{\partial \mathcal{L}_{\text{total}}}{\partial \theta} = \frac{\partial}{\partial \theta} \sum_{i=1}^N \mathcal{L}(f(\mathbf{x}^{(i)}), \mathbf{y}^{(i)}) \quad (3.6)$$

In Equation (3.5), λ is called the **learning rate** and controls the step size of the update. If the learning rate is too small, then the training might converge too slowly; if it is too large, we might immediately observe divergence from the expected output. Training with gradient descent is only guaranteed to find a *local* minimum of f_{θ} , but not a global one².

We now make two observations. First, from Equation (3.5) it arises that in order to perform one update step, we first need to calculate the gradient of the cost on the *entire* training set. This also suggests that the total cost function should be *decomposable* into a sum of individual (per-sample) costs. Because training sets are large and one full pass over the data is computationally expensive, **stochastic gradient descent** is more commonly used and replaces the total cost with a stochastic estimate of it, computed on a randomly sampled

²Such solutions are generally acceptable if the *validation* cost function is sufficiently low.

subset of the data, which we call a **(mini-)batch**. More variations of stochastic gradient descent exist, for example [331, 389] introduce momentum, which acts as an acceleration of gradient steps. Other recent optimizers are adaptive, *i.e.* they find individual learning rates for each parameters, *e.g.* Adagrad [90] and Adam [202].

The second observation is that the computation of the gradient of cost \mathcal{L} with respect to *all* current parameters $\theta^{(t)}$ is necessary. In neural networks and deep learning models, this is achieved with an algorithm known as *back-propagation*.

Back-propagation

To better highlight the fact that the neural network is structured in layers, let us represent $f(\mathbf{x})$ (Equation (3.3)) as a composite function $f(\mathbf{x}) = f^{(2)}(f^{(1)}(\mathbf{x}))$, where $f^{(1)}(\mathbf{x}) = \phi(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)$ is the first layer and $f^{(2)}(\mathbf{x}) = \mathbf{W}_2\mathbf{x} + \mathbf{b}_2$ is the second layer.

Using the chain rule, the partial derivatives of a cost function \mathcal{L} with respect to \mathbf{W}_1 and \mathbf{b}_1 are:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}_1} = \left(\frac{\partial \mathcal{L}}{\partial f^{(2)}} \frac{\partial f^{(2)}}{\partial f^{(1)}} \right) \frac{\partial f^{(1)}}{\partial \mathbf{W}_1}, \quad (3.7)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{b}_1} = \left(\frac{\partial \mathcal{L}}{\partial f^{(2)}} \frac{\partial f^{(2)}}{\partial f^{(1)}} \right) \frac{\partial f^{(1)}}{\partial \mathbf{b}_1}. \quad (3.8)$$

It becomes clear that earlier layers in the network rely on the computation of gradients in later layers, *i.e.* gradients that are computed are propagated backwards. Back-propagation is efficient because a lot of these computations will be shared — *e.g.* it is possible to compute $\frac{\partial \mathcal{L}}{\partial f^{(2)}}$ and then store it and re-use it for all future computations. This extends to deep networks with many layers.

3.2.2 Convolutional Neural Networks

There are several enabling factors that led to the widespread applicability of Convolutional Neural Networks (CNNs) [232] in computer vision.

One reason is that images, as well as other types of data, are high-dimensional representations of raw signals. One major issue with the ANNs mentioned above is the full connectivity of neurons among layers, *i.e.* every neuron is connected to every input unit. Because of this, the number of parameters in the network can grow exponentially; given large input data it can be prohibitive in terms of memory, training samples, required time, and prone to over-fitting.

Additionally, images have inherent structure and local patterns (*e.g.* edges) that repeat over the image. If we distorted the image by shuffling all pixels, it would now be unrecognizable by humans, but it would make no difference for the weights of a fully-connected feed-forward network. Instead, we can take advantage of structured inputs by learning smaller weight matrices which are only locally connected and can be applied over the image in a

sliding window fashion; this gives rise to the properties of *translation equivariance* and *spatial parameter sharing* and can be in practice implemented by a **convolution**.

The input region that a neuron of a given layer “sees”, *i.e.* is connected to, is called **receptive field**. It arises that units in deeper layers will thus have a larger receptive field than units in earlier layers.

After AlexNet, for the next three years, major stepping stones in performance were driven by improved architectures - still on the task of object recognition and the ImageNet challenge. AlexNet was followed by VGG [378], InceptionNet [394] and ResNets [151] which kept on improving over each previous model. These architectures were then commonly used as backbones for other tasks that were thus also improving. Next, we give a brief overview of the most common layers in these modern, deep CNNs.

Convolution

Convolutional layers are the main building blocks of CNNs, as suggested by the name. A convolution applies the dot product between a filter and a local region of the input data, sliding over all spatial locations of the input. If the input is two-dimensional, so is the output. If the input is three-dimensional, then typically a 3D convolution is applied (across channels), thus also resulting in a 2D output—although other variations exist as well. Convolutional filters can be also applied with *stride*, *i.e.* skipping spatial locations, resulting in a reduction in the spatial dimensions.

The filters are usually small kernels (*e.g.* 3×3 spatially) and include learnable weights. At every convolutional layer, we apply several filters, each of them learns to extract a different feature from the input. Thus we usually refer to the output as *feature maps* (one per filter).

As mentioned earlier, activation functions are important to learn complex, non-linear mappings. Convolutions are almost always followed by an activation function and a common choice in CNNs are ReLUs or their variants.

Pooling Layer

Pooling layers are most often used to perform sub-sampling, *i.e.* they are applied with stride, through not necessarily. Typical pooling operations find the *maximum* or *average* with a pre-specified window size, which is usually also small (*e.g.* 3×3), applied on the input data on each channel (feature) independently.

By design, pooling layer do not include any learnable parameters. One reason why we might want to include pooling layers and down-sample features in a network is efficiency, especially when dealing with high-dimensional input data, such as images. One noteworthy property of pooling layers is also translation invariance, as the precise location where features come from inside the pooling window is lost. While this was believed to be useful in problems like image classification, later architectures usually replace most pooling layers with strided convolutions [151].

Dropout

Dropout [158, 382] is a regularization technique which was first used in AlexNet [215]. It reduces the over-fitting of a network to the training data by preventing the co-adaptation of neurons. This is done by setting the activations of a random set of neurons in a layer to zero, thus they do not contribute to the subsequent layers. Dropout is usually inserted right before the prediction layer(s) of a CNN.

Batch Normalization

Several normalization techniques have been proposed for CNNs [22, 215, 296, 363, 414, 448], but in all methods discussed in this dissertation we use batch normalization [179]. The goal of normalization is to facilitate convergence and decrease a network's training time.

[151] shows that it becomes possible to train very deep networks using batch normalization and residual connections. During training, batch normalization computes the mean and standard deviation of each feature across the *batch* and then uses these statistics to normalize the feature to zero mean and unit standard deviation. Then, it introduces two learnable parameters per feature, a scaling factor γ and a shift β , which are applied to the normalized feature. using the statistics of the *batch*.

The application of batch normalization differs at test time due to the lack of population statistics. For this reason, an exponential moving average is updated with the estimated means and standard deviations during training, which are then applied to the features during inference.

Residual Blocks

There have been significant contributions in the design of deep architectures over the years, which aim at improving performance and study the importance of the network width and depth [151, 394] and the connections between layers [151, 167, 172, 453].

Of special interest in this thesis is the concept of residual learning, introduced by He et al. [151]. Residual connections were introduced by [151] to enable better gradient flow inside the network. The idea is to introduce a “shortcut” connection that skips over some layers and sums to their output features of a previous layer. If we denote the desired mapping through some non-linear layers as $h(x)$ — where x is the input to this set of layers, which could come from a previous layer — we have $h(x) + x$ at the output of this building block. This effectively means that we only need to learn a residual mapping. By stacking many such layers, it is possible to build very deep architectures that alleviate the problem of vanishing gradients. Variants with 50, 101, 152 and 1000 layers are proposed in [150].

To account for strided convolutions that change the spatial dimensions of the feature maps inside a residual block, “projection” connections are also introduced and use a 1×1 convolution on the shortcut branch to match the dimensions and make the element-wise sum of features possible. In Chapter 4 we extend this idea to layers that perform feature up-sampling.

3.2.3 Recurrent Neural Networks

Recurrent neural networks (RNNs) are useful for learning problems with sequential data $\mathbf{x} = (x_1, \dots, x_T)$ (for example sentences in language models). An RNN consists of a hidden state \mathbf{h} and an (optional) output \mathbf{y} . At each time step t , the hidden state is updated through a non-linear function g :

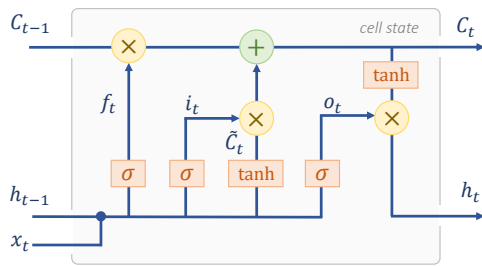
$$\mathbf{h}_t = g(\mathbf{h}_{t-1}, \mathbf{x}_t), \quad (3.9)$$

given the previous hidden state and the current input. A simple RNN “unrolled” over time can be seen in Figure 3.1 (c). In “vanilla” RNNs, g can be very simple, *e.g.* a tanh-activation of the weighted sum of the inputs. Because RNNs are trained with back-propagation *through time*, when modeling long-term dependencies, the gradients which are back-propagated can become very small and *vanish*.

Long Short Term Memory

Long short-term memory units (LSTMs) were introduced by Hochreiter and Schmidhuber [159] and provide a more complex implementation of $g(\cdot)$ using three gates (input i_t , output o_t and forget f_t) and a cell state c_t which represents the memory.

All \mathbf{W} and \mathbf{U} are weight matrices, t is the current time step, σ is the sigmoid activation function and \odot denotes the Hadamard (element-wise) product between vectors. The input gate controls how much information flows into the memory, while the output gate controls how much of the information stored in memory will be used for computing the next hidden state. The forget gate decides what information from the previous cell state will remain the memory of the LSTM. This implementation (also seen in Figure 3.2) tackles the vanishing gradient problem of vanilla RNNs.



$$\begin{aligned} \mathbf{f}_t &= \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1}) \\ \mathbf{i}_t &= \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1}) \\ \mathbf{o}_t &= \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1}) \\ \tilde{\mathbf{C}}_t &= \tanh(\mathbf{W}_C \mathbf{x}_t + \mathbf{U}_C \mathbf{h}_{t-1}) \\ \mathbf{C}_t &= \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{C}}_t \\ \mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{C}_t) \end{aligned} \quad (3.10)$$

Figure 3.2 Equations and schematic of an LSTM.

Gated Recurrent Units

Gated recurrent units (GRUs) [66] are similar to LSTMs but have no output gate and fewer parameters. The internal mechanism of such unit includes a reset gate and an update gate. For the t -th step with hidden state \mathbf{h}_t , reset gate \mathbf{r}_t and update gate \mathbf{z}_t , a GRU functions as follows.

In a GRU, similar to the LSTM, the update gate controls the information flow into the memory. On the other hand, the reset gate controls the information that leaves the memory, *i.e.* if

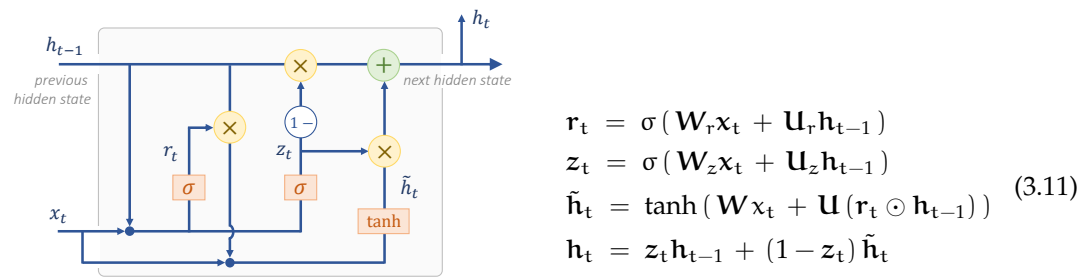


Figure 3.3 Equations and schematic of a GRU.

it is zero, the previous state \mathbf{h}_{t-1} is forgotten and the current state \mathbf{h}_t “resets” to the current input.

3.3 Supervised and Unsupervised Learning

In terms of supervision, machine learning methods can be grouped into two categories: supervised and unsupervised learning. In the earlier example on weather prediction, we were looking at a supervised problem. This means that additionally to the input data \mathbf{x} , we were given labels \mathbf{y} as the expected outcome. Generally, supervision comes through a set of *known* labels (ground truth) that are associated with each data sample. This defines the task we need to learn explicitly, in the sense that given \mathbf{x} we need to learn a mapping to predict the corresponding \mathbf{y} .

In unsupervised learning, we are only given the data \mathbf{x} and no labels. In this case the task is defined differently as it is no longer possible to measure and minimize a cost with respect to expected output \mathbf{y} . The algorithm has to discover properties of the data, learn its probability distribution or group similar elements such as in clustering. Unsupervised learning is attractive since crowd-sourcing or generating labels for data by hand is very time consuming and costly. Thus we want to exploit an abundance of data that is already available and can be easily obtained — usually from the internet.

Currently, hybrid schemes (weakly or semi-supervised learning) are becoming more and more attractive. The promise is that one can learn a good representation of the data with very few or no labels using mostly unsupervised learning techniques. From this learned representation one hopes to reduce the amount of supervision that is necessary to learn a task. Currently, this is not the case — in almost all areas of machine learning, supervised methods are still superior to their weakly or unsupervised counterparts. However, recently more and more progress to match performance has been made.

Finally, the term *self-supervised* learning has become more popular although it lacks a clear definition. As it stands now, self-supervision is a subclass of unsupervised learning, where a proxy-task is derived from the data alone. This usually means removing some aspect from \mathbf{x}

and then learning a function f that recovers this information. The idea is that, if the pre-text task is chosen smartly, f learns a good representation of the data.

3.4 Transfer Learning

Transfer learning is the concept of adapting a machine learning model that has been trained for one task to another task. Maybe the most common occurrence of transfer learning in computer vision is when initializing a classification network [151, 215, 378] with weights pre-trained on ImageNet [360]. The features of a network trained to recognize 1000 different objects from 1.2 million training images are very generic and can then be used as initialization and then *fine-tuned* on other tasks and data such as semantic segmentation, object detection or medical images. It is possible to finetune the entire network or just the deeper layers, thus learning “task-specific” weights.

Transfer learning should not be confused (but sometimes is) with the term *domain adaptation*. While transfer learning is concerned about changing the task, *i.e.* the *output*, of the model, domain adaptation describes the change in the input of the model. For example one has learned depth estimation from synthetic data only; so domain adaptation is needed to make the model work also on real images.

Part II

Scene Understanding through
Geometry and Semantics

4.1	Introduction	25
4.1.1	Motivation	25
4.1.2	Contribution	26
4.2	Related Work	27
4.3	Depth Estimation from a Single Image	31
4.3.1	Fully Connected versus Fully Convolutional	31
4.3.2	Fully Convolutional Residual Network (FCRN)	32
4.3.3	Loss Function	36
4.4	Semantic Segmentation	38
4.4.1	Architecture Modification	39
4.5	Experimental Results	39
4.5.1	Experiments on Depth Estimation	41
4.5.2	Applications of Depth Estimation	47
4.5.3	Experiments on Semantic Segmentation	49
4.6	Conclusion	52

4.1 Introduction

The goal of scene understanding is to study objects not in isolation, but as part of a scene, thus understanding the overall layout and appearance, as well as the spatial and semantic interactions with other objects. In the first part of this dissertation we focus on extracting information about geometry and semantics from single images. This chapter is based on our previously published work on monocular depth estimation [227] and is subsequently also extended to semantic image segmentation.

4.1.1 Motivation

Images are two-dimensional projections of the real three-dimensional world. As such, recovering the underlying scene geometry is a crucial part of understanding. In addition to color, complementary 3D information holds the potential to improve several computer vision tasks, such as reconstruction [306], semantic segmentation [93], human pose estimation [402] or 3D object pose estimation [442]. Depth estimation is a crucial task in scenarios where sensors for direct depth measurements are not applicable or not available. Thus it

comes as no surprise that estimating depth from images has been a well-researched topic in computer vision; related work will be extensively discussed in Section 4.2.

When it comes to humans, 3D perception is linked to binocular cues, *i.e.* cues about the world as seen with both eyes. The underlying principle that allows to perceive 3D structures and which has also inspired computer stereo vision is disparity. Due to the horizontal distance between the eyes of humans and most animals, each eye sees the same scene from a slightly different viewpoint, thus different information about objects' shapes or boundaries is projected into each retina. The two different views are processed by the visual cortex resulting in a 3D model of the environment [26, 121]. Analogously, in computer vision, classic approaches for depth estimation have relied on multiple-view geometry, estimating 3D structures from 2D point correspondences in two or more images via triangulation [149].

On the other hand, depth perception from a *single view*, is an inherently ambiguous problem as it requires inferring a 3D point from a color value alone. As a projection of the 3D world, a 2D image corresponds to infinite 3D scene arrangements, which are, for example, subject to scale. Nonetheless, humans can perceive depth with just one eye using monocular cues, even though depth perception from monocular vision is indeed hindered when compared to binocular vision. Examples of monocular cues include motion parallax, relative sizes of objects or sizes of known objects/landmarks, texture gradients, shading, etc. As our world is made up of known structures, humans can infer sizes and distances reasonably well from previous experience. In most cases, estimating depth from monocular cues is possible up to a scale; however, knowledge of an object's actual size can be used to infer its absolute distance to the eye/camera.

Likewise, computer vision approaches that aim to infer 3D shapes (Shape-from-X) from monocular cues only have traditionally relied on motion [413], shading [480], illumination changes [443] or defocus [393]. It is important to note that although these methods do not make use of multiple views *simultaneously*, they still require multiple images of the same scene. On the other hand, traditional methods to recover depth information from just a single image often necessitate the use of strong geometric priors [163, 195], symmetry [300, 408] or explicit modeling of monocular cues [366].

Most recently, the revolution brought to computer vision by big data and deep learning has resulted in numerous learning-based methods — fully supervised or self-supervised — with unprecedented quality in estimating 3D structure from a single image.

4.1.2 Contribution

In this chapter, we propose a novel CNN architecture to accurately estimate depth from a single image. The method presented in this chapter was published in [227] and the code is publicly available¹ (2016).

¹<https://github.com/iro-cp/FCRN-DepthPrediction>



KEY CONTRIBUTIONS

- We introduce a Fully Convolutional Residual Network (FCRN) that extends the powerful idea of residual connections [150] to up-sampling convolutional operators; this design yields an encoder-decoder architecture based on the successful ResNet architecture [150] which was originally proposed for image classification.
- We show that the proposed architecture consists of fewer learnable weights and requires fewer samples to train than previous work on this task.
- We also investigate different loss functions for the task of depth estimation. We propose to use the reverse Huber norm as an objective function to train the network and explain why it is well-suited for this task.

At the time of publication, our method delivered state-of-the-art results for monocular depth estimation and remains competitive up to date (2020).

The developed architecture is designed to address high dimensional tasks, for example image-to-image translation problems, with efficiency in mind. Therefore, in addition to depth estimation, later in this chapter (Section 4.4), we also present an extension of the same architecture to semantic image segmentation and in Chapter 6 to heatmap-based localization in scenarios applied to robotics.

4.2 Related Work

Traditional methods for estimating depth were geometric and made use of stereo vision [289, 368, 380] which can yield unambiguous reconstructions of 3D structures via triangulation from point correspondences in pairs of images. Despite its ill-posedness, there is also a vast amount of literature related to *single-view* depth estimation. Since this is also the focus of our approach, we will extensively discuss the single-view setting here. In absence of stereo correspondences, earlier methods often rely on geometric assumptions in order to infer the 3D structure of a scene. More recently, there has been an abundance of deep methods that tackle this problem by learning from a large amount of training examples; these have been proven very powerful in monocular depth estimation, pushing the state of the art. This is also partly due to the availability of large-scale, high-quality RGB-D data of unconstrained environments, facilitated by the production of sensors such as Microsoft Kinect or LiDAR.

Shape-from-X

Shape-from-X is a family of methods classified as single-view, meaning that they do not make use of multiple views at the same time. However, these methods use multiple images

of the same scene that might come from different viewpoints, lighting conditions, etc., taken in different points in time. Structure-from-motion (SfM) [104, 413, 444] is arguably the most popular approach. SfM methods can reconstruct a rigid scene from multiple images (taken at various viewpoints of the same scene) and point correspondences among them.

Other environmental assumptions have also been used to infer 3D structure. In shape-from-shading [166, 480] one tries to infer the shape of an object or the geometry of the scene from the observed shading. This often requires either multiple views or multiple images from the same view with different lighting conditions.

Shape-from-symmetry [108, 299, 379, 407] makes use of the information contained in single images of symmetric objects. A key step here is to discover the correspondences between symmetric points on the object. Most work assumes these are given through user input, though in [379] symmetries and correspondences are detected using feature descriptors in the image. Very recently, the shape-from-symmetry has been brought back using deep networks to learn the 3D shape of whole object categories without supervision [446].

Classic Approaches

Classic approaches have used graphical models such as Markov Random Fields (MRF) to address the problem of depth estimation from a single image. In the pioneering work of Saxena et al. [366], depth is estimated by extracting local features representing monocular cues and a MRF is trained to model neighbor dependencies in multiple scales. Their work is later extended to Make3D [365], a method for 3D reconstruction and an accompanying RGB-D dataset that is now a commonly used benchmark for depth estimation. Liu et al. [255] incorporate semantic understanding into depth estimation by using predicted labels to direct the geometric reconstruction. A separate depth estimation model is trained for each semantic category and the predictions are then combined using another MRF. In a more recent approach, Ladicky et al. [225] jointly predict semantics and geometry in a classification approach, *i.e.* predicting the likelihood of a pixel to belong to a certain semantic category and canonical depth.

Instead of inferring absolute depth values, Hoiem et al. [161, 163] propose a statistical method to estimate coarse geometric categories (sky, ground, vertical) depending on the orientation of image regions and use this rough estimation to reconstruct 3D models of outdoor scenery. Subsequently, the same authors focus on occlusion boundaries to improve depth estimation for cluttered scenes [164]. Similarly strong assumptions are made about the structure of indoor environments in [77], that finds floor-wall boundaries to yield a box-like 3D reconstruction; however, this does not hold true for cluttered scenes. These methods are consequently related to estimating scene layout, which is also addressed in contemporary work [140, 154, 165].

Karsch et al. [190] follow a data-driven non-parametric approach to depth estimation. The general direction of non-parametric methods is to search a large RGB-D database for the closest neighbors to a query image and combine their corresponding depth maps to infer the final depth map for the query image. The retrieval of similar RGB images is feature-based, using handcrafted features such as GIST [309]. Karsch et al. [190] then warp the

retrieved images and depth maps using SIFT Flow [256] and globally optimize the final depth map. This approach inspired subsequent work. A computationally more efficient approach is investigated in [211], by replacing SIFT Flow with median filtering over the retrieved depth maps and cross-bilateral filtering on the combined depth. Later on, Liu et al. [258] further improve performance by combining non-parametric methods and graphical models, in particular, optimizing the fusion of the retrieved depth maps with a discrete-continuous graphical model. These retrieval-based approaches rely on the assumption that photometric similarities translate to similar absolute depth values.

Fully Supervised Approaches

In 2014, the seminal work of Eigen et al. [94] introduces a deep learning approach to monocular depth estimation. They propose a coarse-to-fine scheme, which is later also extended to surface normals and semantic segmentation [93]. Our work is inspired by the potential that this method demonstrated and further addresses network efficiency.

Since then, there has been a large body of work that learns depth predictors from single images using RGB-D data. The general idea is to learn a mapping from color images to depth maps, supervised by ground truth depth data. Most of the related work contributes architectural improvements, gradually advancing the state of the art [64, 111, 148, 168, 240, 455]. Some methods focus on the combination of deep architectures and Conditional Random Fields (CRFs), either as a post-processing step [238, 298, 432], operating on CNN features [257] or integrated into the CNN for combining multi-scale information [455, 457]; in [350] shallow networks are combined with random forests. While the majority of methods address depth estimation as a regression problem, others formulate it as a classification [51, 461] or ordinal regression problem [111] or employ adversarial training [58]. To address uncertainty in the context of depth estimation, Kendall and Gal [196] follow a Bayesian approach, while Yang et al. [461] directly predict multi-modal distributions.

Instead of data acquired by depth sensors, some approaches train networks for monocular depth estimation using weakly labeled data or alternative means of supervision. For example, sparsely annotated ordinal relations (DIW) are used for learning relative depth from web data in [61]. Li and Snavely [246] generate ground truth data (MegaDepth) for web photos of landmarks using SfM and multi-view stereo techniques; Chen et al. [60] collect depth information from YouTube videos via SfM and propose a network to judge whether the produced depth quality is acceptable; in [449] disparity maps are automatically generated from stereo pairs on the web (ReDWeb); these datasets are then used to supervised *relative depth* estimation networks. Other use synthetic data and domain adaptation techniques to bridge the gap between synthetic and real data [21, 138, 305].

Unlike relative-depth methods, all of the previously discussed absolute-depth methods train networks with implicit environmental assumptions, using either indoor or outdoor datasets, but not both simultaneously. A recently emerging direction is to develop a single model for both settings. Li et al. [243] tackle this task with an attention mechanism and by recasting the output as depth classification. Facil et al. [100] propose a method that takes known camera intrinsics into account and allows for generalization across different camera models

and datasets. Very recently, Lasinger et al. [231] take a step further and propose a set of loss functions for jointly training a model using diverse datasets and 3D movies in a multi-task learning fashion. At test time, the model is able to transfer to previously unseen datasets.

Furthermore, there exist several methods that jointly learn depth prediction and semantics [181, 185, 197, 298, 432] or explore other multi-task settings [201, 330, 456].

Self-supervised Approaches

In most cases, fully supervised methods require the acquisition of accurate ground truth depth data, which is often challenging or expensive—especially in the case of outdoor scenery. For this reason, learning depth in a self-supervised manner, *i.e.* using only image data, has been a very active field in recent years. In this setting, self-supervision refers to training by image reconstruction and the type of data used is either stereo image pairs, stereo video streams or monocular video streams. In the case of monocular video, due to the lack of a stereo setup, the camera pose between frames is unknown, which makes the problem even less constrained and depth can be only estimated up to scale. Stereo data provides an easier setting also due to the concurrent acquisition of images, which implies a momentarily static scene. In both cases, depth typically arises from a *latent* representation which is used to warp an image and synthesize a nearby view by minimizing the photometric reprojection error.

Stereo Data Some of the first approaches to propose this setup are [452] that uses 3D movies to train a model for disparity estimation, [116] that uses stereo pairs as supervision for learning continuous depth and [126] that further improves performance with a left-right consistency constraint. Further advances include adversarial training [4, 320] or extending to trinocular views [322]. In [224], self-supervised learning is combined with full supervision from sparse depth maps.

Monocular Sequences Zhou et al. [488] propose to use monocular video, treating successive frames in the video sequence as pairs of a (small-baseline) stereo setup and assuming mostly static scenery. In their approach, depth and camera motion are estimated and used to warp input views to the target view, which is used as supervision. Thus, this method is one of the first instances of *deep SfM*, which we will also thoroughly discuss in Chapter 5. Such approaches are largely enabled by the differentiable bilinear sampling module proposed in [180]. Yin and Shi [471] improve upon this setting by decoupling rigid and non-rigid motion via optical flow estimation, thus being able to better account for dynamic objects. Optical flow is also jointly estimated in [63, 266, 336, 492]. To improve the estimated geometry, geometric consistency objectives are proposed in [37, 273] and consistency with surface normals is used in [465], while [319] uses cycle consistency. To address the depth scale ambiguity, [244, 477] use both stereo and temporal information. Using a differentiable visual odometry module, Wang et al. [427] show that it is possible to learn depth without predicting the camera motion.

When training with self-supervision and geometric/photometric constancy objectives, the performance of models is generally lower than when directly regressing depth with super-

vision. To tackle the challenges that arise in the self-supervised setting and bridge the gap to fully supervised performance, several methods have used combined forms of supervision, in particular with dense or sparse depth “ground truth”. One such example is the work by Kuznetsov et al. [224]. Some methods use SfM [208], visual odometry [14, 464] or stereo matching [410] to obtain depth measurements from monocular videos or stereo images and use that as an additional supervisory signal. In [437] depth hints obtained via stereo matching are used in the loss function only under the condition that they reduce the photometric reprojection error. On the other hand, [138, 268, 482] combine self-supervision with synthetic data.

Enabled Applications

Recovering depth and thus scene geometry is a relatively low-level task in the hierarchy of computer vision applications. This means that once it can be carried out sufficiently well, it enables plenty of higher-level tasks that depend on a three-dimensional understanding of the scene, for example in self-driving cars or robotic grasping.

Depth estimation can also be useful in medical applications such as in endoscopy [259, 275] or to aid visually impaired users [31]. Further, one can use the estimated depth map to create compelling visual post-processing effects, such as artificial depth of field effects (defocus) or the Ken Burns effect when animating a camera pan for a still image [308]. Depth estimation can also be used in conjunction with, or to enhance SLAM, as we will discuss extensively in Chapter 5. Since depth sensors, especially laser scanners, like LiDAR, produce very sparse maps, a considerable amount of research has also been devoted to depth map completion/densification [269].

Going one step further, in [82] we extend the 3D understanding of the scene from a single depth map to two depth layers (foreground/background), thus completing objects that are occluded by others. This idea is further extended by Dhama et al. [81] to several object-driven layers. This might enable robots or other agents that need to navigate the scene and understand the 3D layout beyond a single depth map.

4.3 Depth Estimation from a Single Image

Next, we discuss our approach for monocular depth estimation and in Section 4.4 its extension to semantic segmentation. In Section 4.3.2 we propose a deep learning architecture suitable for both tasks, elaborating on the specific design choices and layers. In Section 4.3.3 we discuss the objective function used to train our model for depth estimation.

4.3.1 Fully Connected versus Fully Convolutional

During the time this work was carried out (2016), the most influential CNN architectures were developed for image classification. This essentially implies an encoding nature for

these architectures, *i.e.* mapping input images into feature vectors through repeated convolutions and pooling operations which cause the feature maps to shrink spatially. Stacking convolutional and pooling operations creates deeper models and is necessary for modeling higher levels of abstraction in the features, but also for obtaining sufficiently large receptive fields and consequently allowing the model to capture global context with respect to the input. Typically the head of the network consists of one or more fully-connected layers.

However, this type of architecture is not suitable for dense (per-pixel) prediction tasks. The first work to address depth estimation with deep neural networks [93, 94] employs fully-connected layers and the resulting output is subsequently reshaped to be two-dimensional. In order to increase the resolution of this coarse output, non-learnable upsampling (*e.g.* bilinear) is applied. The upsampled output is fused together with the RGB input followed by additional convolutional layers for further refinement. This approach closely follows “classification” architectures in that it does not get rid of fully-connected layers and consequently vector outputs. However, since the upsampling operation is differentiable the full model (coarse and fine stages) can be trained end-to-end.

There exist few instances of *fully convolutional* architectures prior to our approach. Long et al. [260] directly treat the fully-connected layers of existing architectures [215, 378] as 1×1 convolutions which allows the model to operate on arbitrary input resolutions. This results in small-resolution feature maps, *e.g.* 8×8 , which are then upsampled using *deconvolutional* operators, *i.e.* by reversing the forward and backward passes of a standard convolution. The authors show that upsampling in multiple stages is more effective than a single increase of resolution by an equal amount. Some of the first occurrences of deconvolutional networks is by Zeiler et al. for unsupervised learning in [476] and visualizing already trained classification networks in [475]. Dosovitskiy et al. [88] formulate the upsampling operation as a 2×2 *unpooling*, which doubles the resolution of feature maps, followed by a 5×5 convolution.

4.3.2 Fully Convolutional Residual Network (FCRN)

In contrast to [93, 94], we introduce a fully convolutional network for depth estimation. There are two important aspects to take into consideration. The first one is the effect that the absence of fully connected layers has on the receptive field. The second is to ensure an overall efficient network design, which could allow such models to be used in real-world real-time applications.

Receptive Field

Intuitively, monocular cues that exist in the images hold important information about the geometry of the scene that the model can exploit while learning from a lot of examples. However, most monocular cues appear globally in the image, and not in isolated local regions. Thus, the receptive field, *i.e.* the input region that a given neuron is connected to, becomes an important aspect to consider in monocular depth estimation. In absence of full connections, a full receptive field cannot be guaranteed.

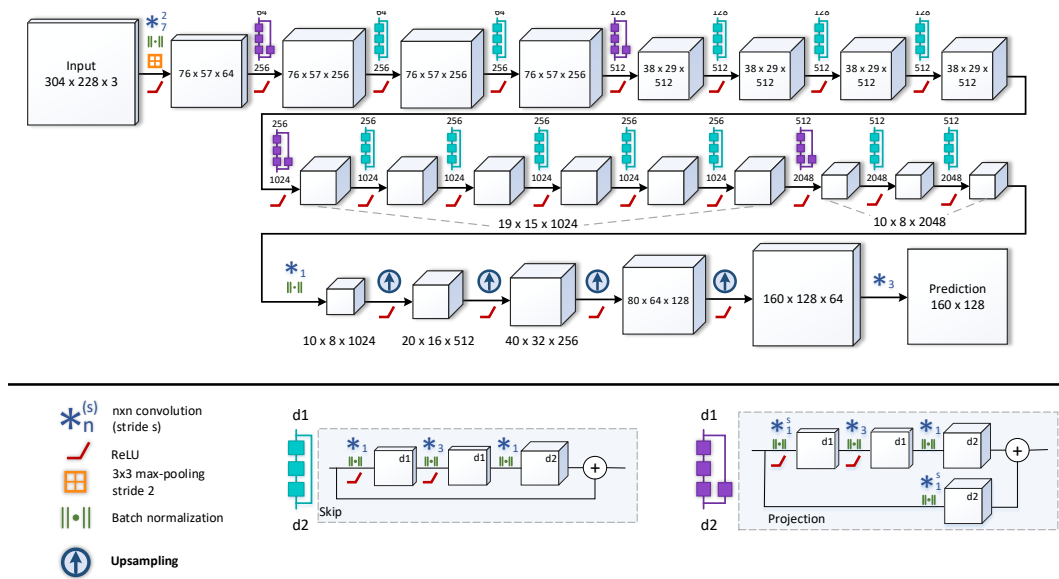



Figure 4.1 FCRN architecture. The architecture extends ResNet-50 to a fully convolutional model. The fully-connected layer is replaced by the proposed up-sampling layers, which result in an output of (approximately) half of the input resolution.

Next, we compute the receptive fields of successful ImageNet architectures, that are typically used in depth estimation through transfer learning. We calculate the receptive fields at the last convolutional layer, before any full connections which would automatically yield a full field of view. The receptive field at the last layer of AlexNet [215] is 151×151 pixels with respect to the input. Similarly, each neuron at the last convolutional layer of VGG-16 [378] is connected to 276×276 input regions. ResNets [150] have opened the way for even deeper architectures, which have — as a by-product — larger receptive fields; ResNet-50 reaches a receptive field of 483×483 pixels.

Eigen et al. [94] train their model with an input size of 304×228 pixels. We adopt the same input resolution for the sake of fair comparisons. Given this input resolution, ResNet-50 achieves a full field of view without the need for fully connected layers, which is a desirable property.

 **NOTE:** *Number of parameters*

Given an input size of 304×228 , the layers with the smallest spatial dimensions (res5a-c) are 10×8 in resolution and consist of 2048 feature channels. As we show later, our fully convolutional model that extends ResNet-50 yields an output resolution of 160×128 pixels. To achieve the same output size through full connections on top of res5c we would require $10 \cdot 8 \cdot 2048 \cdot 160 \cdot 128 \approx 3.355$ billion parameters for this layer alone. Hence, full connections limit the capabilities of deep networks in problems with high dimensional outputs, due to the rapidly increasing number of parameters. This could not only lead to detrimental over-fitting, but can even be prohibitive from a hardware perspective.

Being able to achieve a full field of view with fully convolutional models is important for the task of depth estimation, as well as for other dense tasks that require reasoning at global context. However, it is difficult to study the effect of the receptive field on this task *in isolation*, as it cannot be easily decoupled from the higher discriminative power that each new architecture is equipped with (ResNet > VGG > AlexNet).

Architecture

We build our model as a natural extension of ResNets to fully convolutional architectures, by transferring the idea of residual connections to upsampling operations. The resulting model is an encoder-decoder or U-shaped architecture that consists of residual blocks. We hereby refer to this architecture as a **fully convolutional residual network (FCRN)**. A detailed representation of FCRN can be seen in Figure 4.1. The size of each feature volume that is denoted in the figure is the result of the preceding operations given spatial input dimensions of 304×228 pixels. The encoding part of the architecture is based on ResNet-50 and initialized with weights pre-trained on ImageNet [360]. Our contribution lies in the decoding part, which consists of successive up-sampling layers that progressively increase the spatial dimensions of the feature maps and finally yield a dense output. Next, we explain the design of the decoding part in detail.

Up-Projection Layer

As previously discussed, deconvolution performs the reverse operation of a convolution and is often used for increasing the spatial dimensions of feature maps with learnable parameters. We adopt the formulation of Dosovitskiy et al. [88], where the upsampling operation is described as a 2×2 unpooling followed by a convolution with 5×5 kernels and ReLU activation. We refer to this operation as up-convolution; it can be thought of as the reverse operation of convolution and pooling which is typically used in the encoding part of architectures.



NOTE: Unpooling

In $s \times s$ unpooling, the resolution of the feature maps is increased s times by interleaving the elements of the input with $s - 1$ zeros. When $s = 2$, the size of the output is double in resolution. In [88] each element of the input is placed in the top left corner of a $s \times s$ zero matrix, while in [475] “switches” from the max-pooling operations of the encoder are used to place the elements in the corresponding cell during upsampling.

We extend this idea with residual connections for seamless integration with ResNets [150]. We refer to this new layer as *up-projection*, in consistency with the *projection* connection introduced by He et al. [151], which can be seen in Figure 4.1. We thus follow the up-sampling operation with a 3×3 convolution and introduce a second branch with another up-convolution. Its output is summed element-wise with the output of the main branch. As the outputs need to be of the same size (both should be upsampled), the unpooling operation is performed prior to the residual layer, but the subsequent 5×5 convolutions split into the different

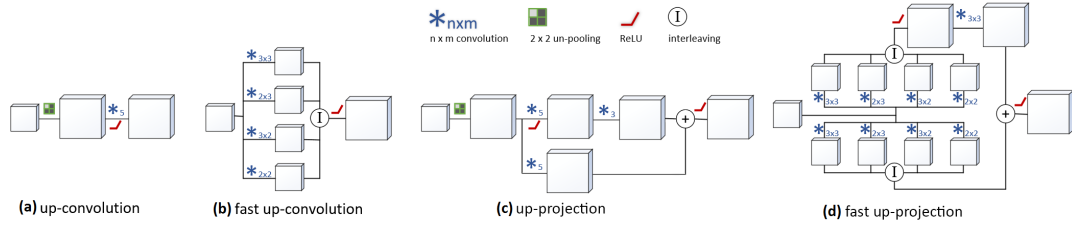


Figure 4.2 Comparison of upsampling blocks. (a) Standard up-convolution. (b) Faster up-convolution. (c) The proposed up-projection. (d) Faster up-projection.

branches and do not share weights. The up-convolution and up-projection layers are shown in Figure 4.2 (a) and (c) respectively.

By design, each up-projection layer doubles the input resolution. With every up-sampling layer, we halve the number of channels in the output. We stack four such blocks, resulting in 16 times upscaling, while the encoding part of our architecture (ResNet) has a total stride of 32. Assuming an input image of dimensions $W \times H$, the bottleneck layer would have a resolution of $W/32 \times H/32$ and, accordingly, the output $W/2 \times H/2$. As an aside, the input dimensions that we use as an example in Figure 4.1 do not divide 32 exactly, thus the output resolution is not exactly half. We did not observe a gain in performance when up-convolving to $W \times H$ with a fifth layer, so we omit that in favor of computational efficiency; the last scale up can be achieved with bilinear upsampling instead.

Faster Up-Convolutions

We further propose a computational modification to the up-convolutional layer, which results in approximately 15% global speed-up during training; this modification is naturally extended to up-projections as well.

Our main observation is that 2×2 unpooling yields feature maps that consist of 75% zeros (more for $s > 2$), which causes a lot of wasteful operations for the subsequent convolution. In Figure 4.3, we present an alternative computation which is mathematically identical to the up-convolution with 2×2 unpooling, 5×5 convolution. Instead of convolving the upsampled feature map with the 5×5 kernels, we perform a set of “smaller” convolutions on the lower resolution map (without unpooling). The kernel sizes are decided by the non-overlapping pixel groups and offsets $\delta = (\delta_x, \delta_y)$ from the top left corner: (A) 3×3 , $\delta = (0, 0)$, (B) 2×3 , $\delta = (1, 0)$, (C) 3×2 , $\delta = (0, 1)$, (D) 2×2 , $\delta = (1, 1)$. These are denoted with different colors in Figure 4.3. We obtain exactly the same output as with the up-convolution, by *feature interleaving*, i.e. spatially alternating the elements of all resulting feature maps, as shown in the same figure. The faster variants of the up-convolution and up-projection layers are shown in Figure 4.2 (b) and (d) respectively.

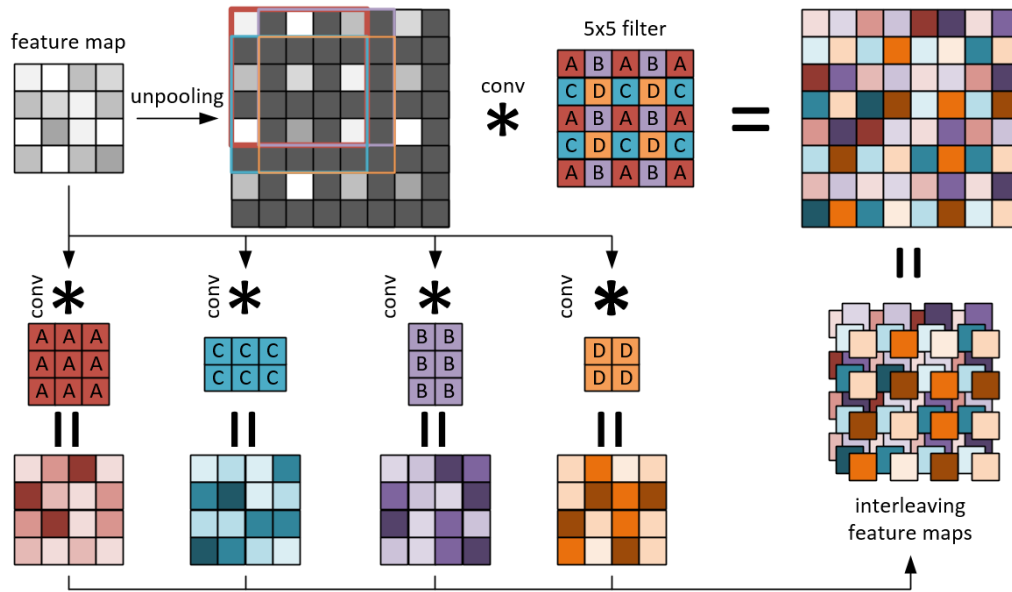


Figure 4.3 Faster up-convolution. Top part: 2×2 unpooling, followed by a 5×5 convolution, resulting in a feature map which is doubled in width and height. In a sliding window fashion, depending on the position of the filter, there are four unique constellations (A,B,C,D), ignoring zero values. Bottom part: one could convolve the low-resolution feature map with the four distinct filters—their shape is decided by the non-zero pixel groups of the high-resolution feature map. It is then possible to interleave the elements of the resulting feature maps to reach an equivalent outcome as the top part, while avoiding zero multiplications. Note: A,B,C,D in the kernels only indicate pixel groups; the actual weight values are not uniform.


4.3.3 Loss Function

Commonly used for optimization in supervised regression problems is the \mathcal{L}_2 loss function:

$$\mathcal{L}_2(y, \hat{y}) = \|\hat{y} - y\|_2^2 = \sum_{i=1}^N (\hat{y}_i - y_i)^2, \tag{4.1}$$

which computes the squared distance between prediction $\hat{y} \in \mathbb{R}^N$ and ground truth data $y \in \mathbb{R}^N$ (with N being the number of pixels/elements in y).

However, the minimization of \mathcal{L}_2 assumes (optimally) data that follows a Gaussian distribution. As one can see in Figure 4.4, this is not the case for the depth value distributions of common RGB-D benchmarks, which are heavy-tailed.

 **NOTE:** *Depth sensing*

Far-ranged depth values are not only scarce but also less reliable when pushing towards the maximum working range of sensors, such as Kinect [199]. Kinect v2 has a maximal sensing distance of 8 meters, but works optimally between 1.5 and 4.5 meters.

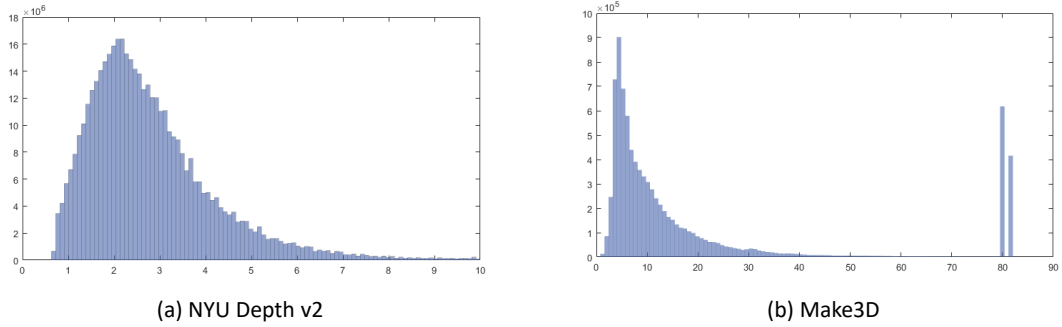


Figure 4.4 Depth distribution in common benchmarks (a) NYU Depth v2 and (b) Make3D. We plot the histograms of discretized depth values. The x -axis indicates depth in meters. In (b) the two spikes around 80m correspond to sky pixels.

Hence, in depth estimation \mathcal{L}_2 might not be the optimal choice. Eigen et al. [94] train their model with \mathcal{L}_2 but overcome this issue by predicting log-depth instead, effectively transforming a log-normal distribution to Gaussian.

Instead, we propose to use the reverse Huber function, which is also known as Berhu. In [312, 493], Berhu has been used as a regularizer in combination with the robust Huber norm as objective. Here, we use it as the objective function, thus minimizing $\mathcal{L}_B(y, \tilde{y}) = \sum_{i=1}^N \mathcal{B}(\tilde{y}_i, y_i)$ with:

$$\mathcal{B}(\tilde{y}_i, y_i) = \begin{cases} |\tilde{y}_i - y_i| & \text{if } |\tilde{y}_i - y_i| \leq \alpha, \\ \frac{(\tilde{y}_i - y_i)^2 + \alpha^2}{2\alpha} & \text{otherwise.} \end{cases} \quad (4.2)$$

According to Equation (4.2), the loss amounts to the \mathcal{L}_1 norm when the absolute difference between prediction and ground truth is smaller than some constant α and switches to quadratic when the difference is large, lying outside the range $[-\alpha, \alpha]$. Equation (4.2) is continuous and first-order differentiable at α , where the transition between the \mathcal{L}_1 norm and the \mathcal{L}_2 norm happens. Figure 4.5 shows a plot comparing these three functions.

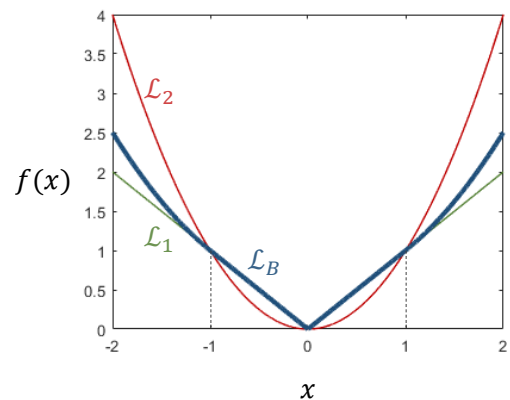


Figure 4.5 The Berhu function \mathcal{L}_B for $\alpha = 1$ in comparison to \mathcal{L}_2 and \mathcal{L}_1 .

We set $\alpha = 0.2 \cdot \max_i |\tilde{y}_i - y_i|$, where i indexes all pixels with a valid depth in the batch. Thus, at each training step we *adaptively* set the constant α to penalize values that are closer to the ground truth with \mathcal{L}_1 and larger errors with \mathcal{L}_2 , according to the maximum error that is observed in the current step.

While Berhu behaves as \mathcal{L}_2 for higher residuals, it increases, at the same time, the back-propagated costs of smaller residuals with \mathcal{L}_1 . In other words, predictions that are already very close to the ground truth depth value are encouraged towards zero with a larger weight, that is, in Figure 4.5, when α is below one meter. This manifests itself more in image regions at a closer range, since then the residuals will also be smaller — for example it is much more likely to have an error of 0.1m when the target object surface lies at 2m than 8m — which coincides with the density of the distributions in Figure 4.4. It is important to note that the loss is *not* a robust loss, *i.e.* it does not ignore outliers (in this case, long ranges); this is the effect that \mathcal{L}_1 alone would have on large residuals. Empirically, we observe that at the beginning of training the constant α is around 2m and quickly reduces to 1m by the end of the first epoch. At the same time, the percentage of pixels for which $|\tilde{y}_i - y_i| \leq \alpha$ begins at 50% but increases to approximately 80 – 85% at the later stages of training. Thus, the majority of small residuals will be penalized with \mathcal{L}_1 , however for the large residuals a stronger \mathcal{L}_2 gradient will be applied.

4.4 Semantic Segmentation

As discussed earlier in the introduction, holistic understanding of a scene requires the integration of different types of information that can be extracted from an image, both geometric and semantic. This includes, for example, recognizing different object categories which are present in the scene, as well as their position, orientation, interactions, etc.

Until now, we have discussed the extraction of 3D information from a single image, thus addressing the *geometric* component of scene understanding. The estimation of metric depth provides information about how different objects are arranged spatially with respect to each other but also information about the scene layout as a whole. However, the notion of *objects* here is still missing; using depth information alone we cannot accurately distinguish scene elements from each other or even identify them based on functional properties or other visual attributes. Thus, in this section we extend the proposed architecture to the task of semantic segmentation, *i.e.* assigning each pixel of the image to a semantic label from a pre-defined set of categories.

NOTE: Common semantic categories

The abstraction of class labels can vary. For example in indoor datasets, commonly annotated classes include *wall*, *floor*, *door*, etc., but given a finer semantic detail it is also possible to encounter classes such as *pillow*, *book*, *clothes*, *i.e.* smaller objects that are found in a household. Often many objects might look similar although assigned to different categories — for example *dining table* and *desk* — or overlapping — for example *blinds* and *window*.

No matter the level of abstraction, semantic segmentation of the scene delineates detailed boundaries of objects as well as classifying them, which is useful in many practical scenarios in the context of robotics or augmented reality.

4.4.1 Architecture Modification

When adapting the architecture which we have presented in Section 4.3 to semantic segmentation, only slight modifications are necessary. Since semantic segmentation can be thought of as pixel-wise classification, the number of channels in the output and therefore the number of the convolutional filters in the very last layer needs to change to match the number of semantic categories C , that is $\tilde{y} \in \mathbb{R}^{\frac{W}{2} \times \frac{H}{2} \times C}$. To convert the raw output into a probability distribution over class labels we use $\text{softmax}(\cdot) : \mathbb{R}^C \rightarrow \mathbb{R}^C$:

$$\text{softmax}(\tilde{y}_i)_c = \frac{\exp \tilde{y}_{i,c}}{\sum_j \exp \tilde{y}_{j,c}}, \quad (4.3)$$

where $c = 1, \dots, C$ indexes the channels of the output. Now, each pixel i of the output can be interpreted as a probability vector, with all its values lying between 0 and 1 and all vector values summing to 1.

The architecture can be trained by minimizing the per-pixel cross-entropy loss between the ground truth and the predicted distributions, which is equivalent to minimizing the negative log-likelihood:

$$\mathcal{L}_{CE} = -\frac{1}{N} \sum_i \sum_c y_{i,c} \log(\text{softmax}(\tilde{y}_i)_c). \quad (4.4)$$



NOTE: *Softmax cross-entropy*

The use of the exponential function in softmax is convenient when training with log-likelihood. In practice, in current deep learning frameworks, instead of Equation (4.3) and Equation (4.4), the softmax function and log-likelihood calculation are merged in a more numerically stable formulation.

4.5 Experimental Results

We experimentally evaluate the proposed architecture on two benchmark datasets, NYU Depth v2 [306] and Make3D [365]. In this section, we present an ablation study with respect to architectural components and objective functions. We then compare our method to prior work in monocular depth estimation and semantic segmentation of indoor scenes and showcase how predicted depth maps can replace range sensors in different applications.

Datasets

We experiment with datasets of both indoor and outdoor scenes (but train separate models for the two cases).

NYU Depth v2 The raw NYU Depth v2 dataset consists of a total of 464 RGB-D sequences of indoor spaces, which are captured with a Microsoft Kinect. There exist 249 training and

215 test scenes. When training the depth prediction model, we only need single frames (no temporal information). We pre-process the data using the toolkit that is provided with the dataset to synchronize and align the image and depth data. Because subsequent frames are very similar, we sample every tenth frame resulting in a subset of 12,000 RGB-D pairs that we use for training the network for depth estimation.

Additionally, there exists a small subset of NYU Depth v2 which consists of 1449 images. This subset contains depth maps where previously invalid depth regions (*e.g.* object boundaries or across specular surfaces and windows) are filled in using a colorization technique [236]. The subset is split into 795 training and 654 testing images. The latter is the set commonly used among depth estimation methods for reporting and comparing performance on NYU Depth v2.

Further, the aforementioned subset is annotated with dense semantic labels spanning 800 object categories. Thus, we use this small dataset for training and evaluating the semantic segmentation network as well.

Make3D A commonly used dataset for outdoor scene depth estimation is Make3D [365]. This dataset consists of 400 training and 134 testing images which are acquired using a custom 3D scanner. The RGB images have a resolution of 1704×2272 , however, due to the capacity of the 3D scanner, the corresponding depth data have a limited resolution of 305×55 pixels.²

Error Metrics

Before moving on to the experiments, we introduce the evaluation metrics that are used in the remaining of this section. We compute the following metrics:

- Absolute relative error (rel): $\frac{1}{T} \sum_i \frac{|y_i - \tilde{y}_i|}{y_i}$
- Root mean square error (rmse): $\sqrt{\frac{1}{T} \sum_i (y_i - \tilde{y}_i)^2}$
- Mean log-error (\log_{10}): $\frac{1}{T} \sum_i |\log_{10}(y_i) - \log_{10}(\tilde{y}_i)|$
- Root mean square log-error (rmse(log)): $\sqrt{\frac{1}{T} \sum_i (\log_{10}(y_i) - \log_{10}(\tilde{y}_i))^2}$
- Accuracy based on a threshold δ_j , that is the % of \tilde{y}_i such that: $\max\left(\frac{y_i}{\tilde{y}_i}, \frac{\tilde{y}_i}{y_i}\right) < \delta_j$ with $\delta_j = 1.25^j$ and $j \in \{1, 2, 3\}$.

T is the total number of (valid) ground truth depth values in a test set.

²More recently, due to the limitations of the depth sensor used for acquisition in Make3D, most methods have turned to KITTI [117] for training and only use Make3D for evaluating generalization.

4.5.1 Experiments on Depth Estimation

Next, we report the performance of several models trained for the task of depth estimation on the two datasets.

Implementation Details

In all cases, the encoder of the architecture is based on an existing classification network (AlexNet [215], VGG [378], ResNet [151]) and initialized with the weights of the corresponding model pre-trained on the ImageNet classification challenge data (ILSVRC) [360]. All new layers are instead initialized with weights sampled from a Gaussian distribution of zero mean and variance equal to 0.01.

We use offline data augmentations to artificially increase the number of training samples, *i.e.* we create a fixed set of randomly transformed samples and use this for training the models. We augment NYU Depth v2 from 12,000 to approximately 95,000 samples (8 transformations per sample) and Make3D from 400 to approximately 15,000 samples. For NYU Depth v2 we first downscale all data to 320×240 pixels and then take a center crop equal to network's input resolution. Then, we use the transformations proposed in [94]:

- Rotating in-plane by a random angle of $[-5, 5]$ degrees.
- Scaling with a factor of $[1, 1.5]$. Scaling is equivalent to a zooming effect, *i.e.* when scaling by a factor the depth ground truth must be divided by the same factor, making the scene “move closer” (for $s > 1$).
- After scaling and rotation, we randomly crop a patch that matches the input resolution (304×228 for NYU Depth v2).
- Color augmentations with a multiplicative factor in $[0.8, 1.2]$ randomly sampled for each channel.
- Horizontal flipping with 50% probability.

All augmentations (except color augmentations) are applied to both the input image and the depth ground truth.

NOTE: *Preserving scene geometry*

Unless random cropping is done symmetrically around the center of the image, it does not preserve the scene geometry (as it effectively changes the camera), though Eigen et al. [94] observed that small augmentations benefit the network training overall. This is why we make crops as large as possible, thus retraining most of the image after the crop while allowing only small translations. If, instead, we allowed large scale factors (over 1.5) cropping patches of 304×228 would give image regions corresponding to only a small portion of the original image and, if the crop is further away from the image center, the associated depth values would be quite off.

We use the same augmentations for Make3D, except for the random cropping. Prior to augmentation we resize all data to 345×460 , as originally suggested by Liu et al. [258],

Architecture		Loss	#params	rel	rmse	\log_{10}	δ_1	δ_2	δ_3
AlexNet	FC	\mathcal{L}_2	104.4×10^6	0.209	0.845	0.090	0.586	0.869	0.967
		\mathcal{L}_B		0.207	0.842	0.091	0.581	0.872	0.969
	UpConv	\mathcal{L}_2	6.3×10^6	0.218	0.853	0.094	0.576	0.855	0.957
		\mathcal{L}_B		0.215	0.855	0.094	0.574	0.855	0.958
VGG-16	FC	\mathcal{L}_2	113.8×10^6	0.288	0.132	0.105	0.362	0.658	0.832
		\mathcal{L}_B		0.180	0.731	0.078	0.659	0.912	0.980
	UpConv	\mathcal{L}_2	18.5×10^6	0.194	0.746	0.083	0.626	0.894	0.974
		\mathcal{L}_B		0.194	0.790	0.083	0.629	0.889	0.971
ResNet-50	FC	\mathcal{L}_B	359.1×10^6	0.181	0.784	0.080	0.649	0.894	0.971
	FC (64×48)	\mathcal{L}_B	73.9×10^6	0.154	0.679	0.066	0.754	0.938	0.984
	UpConv	\mathcal{L}_2	43.1×10^6	0.139	0.606	0.061	0.778	0.944	0.985
		\mathcal{L}_B		0.132	0.604	0.058	0.789	0.946	0.986
	UpProj	\mathcal{L}_2	63.6×10^6	0.138	0.592	0.060	0.785	0.952	0.987
		\mathcal{L}_B		0.127	0.573	0.055	0.811	0.953	0.988

Table 4.1 Comparison of different architectures on NYU Depth v2. We compare different encoders (AlexNet [215], VGG-16 [378], ResNet-50 [151]), up-sampling strategies (FC, UpConv, UpProj) and loss functions ($\mathcal{L}_2, \mathcal{L}_B$). For the reported errors rel, rmse, \log_{10} lower is better, whereas for the accuracies $\delta_j < 1.25^j$ higher is better.

which is the resolution we also use for evaluation. However, as input to the network, we downsample to half resolution, due to hardware considerations.

In all cases, we use bilinear upsampling to resize the output of the network to the resolution used for evaluation by previous methods.

Training Details

We train separate models for indoor and outdoor scenes. In both cases, we optimize using stochastic gradient descent with momentum of 0.9 in batches of 16 samples. The starting learning rate is 0.01, which we decay when the validation error plateaus. We also apply weight decay (\mathcal{L}_2 regularization) equal to 0.0005 and dropout (in the penultimate layer) with probability 0.5. We train on NYU Depth v2 for 25 epochs and on Make3D for 30 epochs, but select the model at the epoch with the best validation error. During training we only compute and back-propagate the cost for *valid* depth pixels, where $y > 0$.

Ablation Experiments

In Table 4.1 we study the effect of the encoder and decoder of the architecture as well as the choice of the objective function on the overall performance. As the encoder we compare three architectures: AlexNet [215], VGG-16 [378] and ResNet-50 [151]. As the decoder (which is the main contribution of this work), we compare for each base architecture a variant using fully connected layers (FC) against progressive upsampling with up-convolutions

(UpConv). Finally, for the ResNet-based models, we also evaluate our fully convolutional residual variant using the proposed up-projection blocks (UpProj).

Encoders It is easy to notice that, generally, with improved network design and increased classification accuracy on ImageNet (ResNet > VGG > AlexNet) comes improved performance on depth estimation as well. Thus, all ResNet variants perform better than the previous models.



NOTE: *How do ImageNet architectures transfer to other tasks?*

Better ImageNet architectures also yield superior performance when transferred (fine-tuned partly or in whole) to other tasks: this has been a common observation across various tasks in the computer vision literature during the last few years. It was not, however, until very recently that it was systematically studied by Kornblith et al. [213], although the evaluation was only done for classification tasks.

As an aside, we have also experimented with randomly initialized weights and found that ImageNet-trained features are useful and constitute an important starting point for depth estimation, especially in smaller datasets. All architectures (especially deeper ones) exhibit very slow convergence and a significant drop in performance when trained from scratch with random initialization.

Decoders Furthermore, we evaluate the effect of the decoder on both the performance and the number of parameters. In Table 4.1, unless otherwise specified, FC layers are trained with the same output resolution as the fully convolutional variant.

For AlexNet, FC layers result in slightly better performance than up-convolutions, which could be accredited to the limited receptive field of this model, as discussed in Section 4.3.2. Intuitively, this does not allow the model to exploit global information. However, when combined with VGG-16 or ResNet-50, our up-sampling layers show a dramatic improvement over the fully connected variants. The residual connections in UpProj yield a further improvement in performance in comparison to UpConv.

We also experiment with two fully-connected variants on top of ResNet-50, one of which yields a lower output resolution of 64×48 pixels. We also observe that this lower-resolution model is better by a significant margin — which demonstrates the difficulty of training FC layers as the resolution increases. In addition to the performance gain, we note the significant reduction of network parameters when replacing fully connected layers with the fully convolutional setup. We can now obtain a high output resolution while using only small convolutional filters, which is necessary for scalability in several image-to-image translation problems. This is an important consideration given hardware limitations (*e.g.* limited GPU memory). Another reason is that models with a large number of parameters are much more difficult to optimize with limited data and are prone to over-fitting (as suggested by the performance drop).

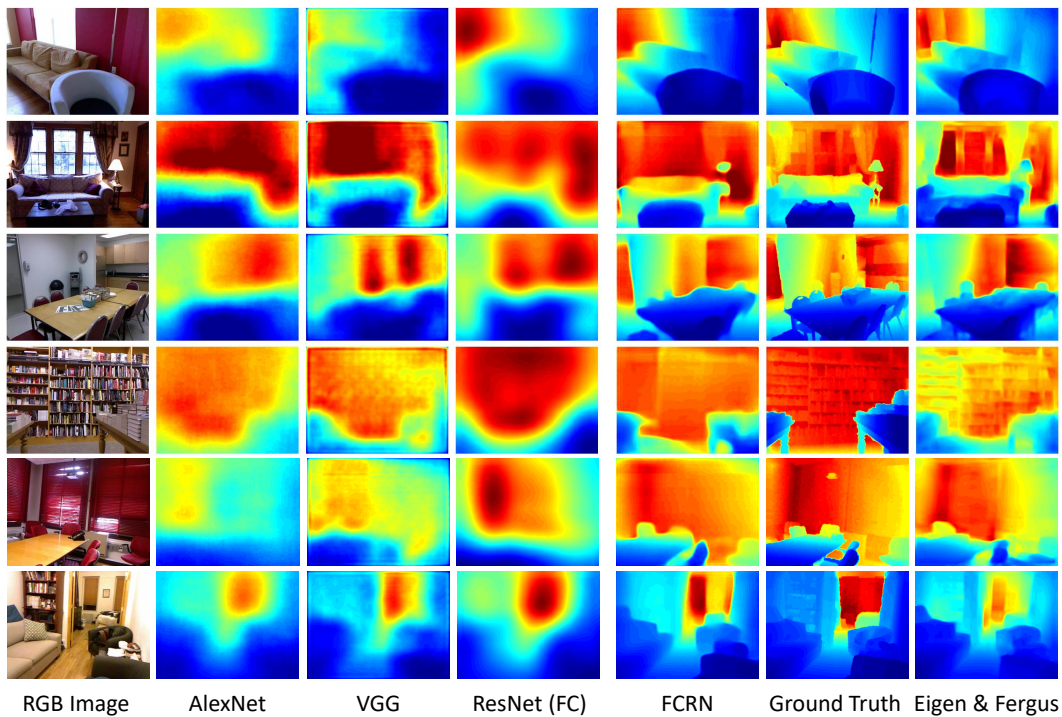


Figure 4.6 Depth predictions on NYU Depth v2. We compare the qualitative results among baselines AlexNet (UpConv), VGG-16 (UpConv), ResNet-50 (FC), and the proposed FCRN. We further compare to the publicly available predictions of Eigen and Fergus [93]. In the colormap, blue is near, while red is far. For better comparisons, all predictions are shown with respect to the ground truth colormap, thus respecting the real scale.

Loss function We train all models with two different objectives. In all experiments in Table 4.1, the Berhu loss \mathcal{L}_B results in better performance than \mathcal{L}_2 and the difference in relative error is bigger. The relative error also puts more weight on closer-range pixels than longer-range pixels, thus the improvement due to the choice of the loss function is more prominent here.

Computation time We also compare the processing time of a single up-convolutional block (1.5 ms) with our up-projection (0.14 ms), during inference on a single image. The speed up is approximately $10\times$. This is higher than the theoretical speed up of $4\times$, which is likely due to using smaller filter sizes as well. Overall, one full forward pass on a single image takes 55ms with the proposed speed up and 78ms with up-convolutions. These experiments were carried out on a single GPU, NVIDIA GeForce GTX TITAN X (Maxwell architecture). Real-time processing speed is a major enabling factor for 3D reconstruction and SLAM, which we discuss in the following chapter.

Comparisons to Prior Work (NYU Depth v2)

We compare the performance of our best model (FCRN) with previous work in Table 4.2, setting a new state of the art. It is important to note that previous work often trains on different amounts of data. For example, [93, 94] train on the full NYU Depth v2, while the

NYU Depth v2	rel	rmse	rmse(log)	\log_{10}	δ_1	δ_2	δ_3
Karsch <i>et al.</i> [190]	0.374	1.12	-	0.134	-	-	-
Ladicky <i>et al.</i> [225]	-	-	-	-	0.542	0.829	0.941
Liu <i>et al.</i> [258]	0.335	1.06	-	0.127	-	-	-
Li <i>et al.</i> [238]	0.232	0.821	-	0.094	0.621	0.886	0.968
Liu <i>et al.</i> [257]	0.230	0.824	-	0.095	0.614	0.883	0.971
Wang <i>et al.</i> [432]	0.220	0.745	0.262	0.094	0.605	0.890	0.970
Eigen <i>et al.</i> [94]	0.215	0.907	0.285	-	0.611	0.887	0.971
Roy and Todorovic [350]	0.187	0.744	-	0.078	-	-	-
Eigen and Fergus [93]	0.158	0.641	0.214	-	0.769	0.950	0.988
ours (FCRN)	0.127	0.573	0.195	0.055	0.811	0.953	0.988

Table 4.2 Comparison to prior work on the NYU Depth v2 dataset. For the error metrics (rel, rmse) lower is better, while for the accuracies δ_j higher is better.

CRF-based methods [238, 257, 432] create a lot of smaller training patches from the labeled subset. We have calculated the number of parameters in Eigen and Fergus [93] to be 218 million (3.5 times more than our model), which further explains the need for the increased amount of data.

In Figure 4.6 we also qualitatively compare the global accuracy and overall structure of the predictions of various models. We show the fully convolutional variants of AlexNet and VGG-16, as well as the FC-variant of ResNet. According to Table 4.1, ResNet-FC significantly improves global accuracy over AlexNet and VGG-16; however, it completely lacks structure, which FCRN is able to recover successfully. In the same figure, we also compare to the publicly available predictions of [93], which is the closest method to our work. Since they refine the initial coarse depth maps in a multi-scale architecture, they are also able to recover image structures sufficiently well, however, sometimes high-textured image regions can cause unwanted depth artifacts (*e.g.* fourth row). This is because the RGB image is concatenated to the initial depth prediction for further refinement.

To summarize, we show that fully convolutional encoder-decoder architectures can tackle the problem of coarseness in depth prediction simply and efficiently, without the need for CRFs or multi-scale architectures, as seen in previous work.

Comparisons to Prior Work (Make3D)

From Figure 4.4 it is easy to see that Make3D samples include several depth values of around 80 meters, which mostly correspond to sky pixels. Because of these inaccuracies, we train and evaluate FCRN on Make3D for pixels that are assigned a depth value up to 70 meters, as suggested by [258]—sky pixels are detected and excluded during training. During training we mask the loss where $y > 70$ to avoid back-propagating errors.

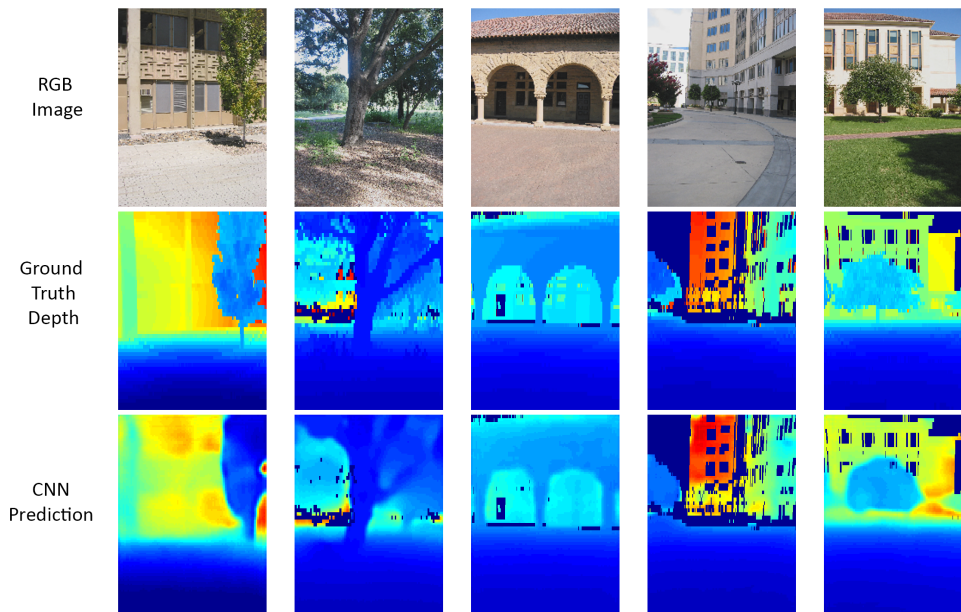


Figure 4.7 **Depth Prediction on Make3D.** Displayed are RGB images (first row), ground truth depth maps (middle row) and our predictions (last row). Pixels that correspond to distances $> 70\text{m}$ in the ground truth are masked out

Make3D	rel	rms	\log_{10}
Karsch <i>et al.</i> [190]	0.355	9.20	0.127
Liu <i>et al.</i> [258]	0.335	9.49	0.137
Liu <i>et al.</i> [257]	0.314	8.60	0.119
Li <i>et al.</i> [238]	0.278	7.19	0.092
FCRN (\mathcal{L}_2)	0.223	4.89	0.089
FCRN (\mathcal{L}_B)	0.176	4.46	0.072

Table 4.3 **Comparison with prior work on Make3D.** We report the performance of our models trained with \mathcal{L}_2 and \mathcal{L}_B losses.

In Table 4.3, we report our results under the C1 criterion ($y \leq 70$) and compare to previous work. In this dataset, the performance gain of \mathcal{L}_B over \mathcal{L}_2 -training is even more prominent. Finally, we show qualitative results of our network on Make3D in Figure 4.7.

Have We Learned Pure Geometry?

Besides the quantitative evaluations it is interesting to develop a deeper understanding about what the depth prediction model has learned. We have extensively experimented with our NYU-Depth-v2 model across various indoor settings. We have observed strong generalization capabilities, using standard web cameras (*i.e.* not the same sensor the model is trained with). The model performs surprisingly well, even in scenes that are not similar to those used in training, for example scenes in which foreground objects are much closer to

the camera or large spaces, *e.g.* auditoria, that extend beyond the typical maximum range of the dataset (8–10 meters).

One question that arises is what makes the model generalize so well. A possible explanation is that, although trained to predict geometry, the model learns representations that extend beyond pure geometry. Indeed, if we vertically flip the camera, thus recording the same scene upside down, the model completely fails to estimate the depth map of the scene — in fact, the prediction does not even resemble the *structure* of the scene anymore, regardless of its absolute value. A model with purely geometric understanding would not face this challenge. This possibly suggests that the model, instead, learns depth through semantics, for example, by recognizing the floor and walls where objects usually lie on or against, as well as the typical appearance and sizes and of common household objects (desks, sofas, beds, etc.) with respect to their distance from the camera. When the viewpoint changes drastically and these structures cannot be accurately recognized, depth prediction also fails. Thus, we believe that inherent to the model is a strong relation between geometry and semantics. This is reminiscent of monocular cues in human perception, for example inferring depth from known object sizes.

The Dollhouse Effect

A general limitation in monocular depth estimation is the implicit assumption of the scene types used to train the algorithm. We have trained one model that specializes in indoor scenes and one in outdoor scenes. Although more recently, some methods attempt to address this limitation by using joint data sources [100, 231, 243], some challenges still persist. For example, if one takes a realistic photo inside a dollhouse with miniature figures, there is no way for current systems to correctly infer the absolute scale, even if relative distances between objects are correct. Nevertheless, such a scenario would play tricks on the human brain as well, and is in fact often used in movies. Although this challenge has not been addressed, it is unlikely to cause a problem in practical real-world scenarios.

4.5.2 Applications of Depth Estimation

We now show some applications where estimated depth can be useful to better understand the advantages and limitations of learned depth prediction as a replacement for depth sensing.

Application to SLAM

Perhaps one of the most important applications of estimating geometry is 3D reconstruction of the environment. Here we briefly examine whether the depth predictions provided by our model are accurate enough to be successfully used inside existing frameworks for RGB-D Simultaneous Localization and Mapping (SLAM)³. The end-goal is dense scene reconstruction in 3D.

³A more comprehensive review of SLAM literature follows in Chapter 5.

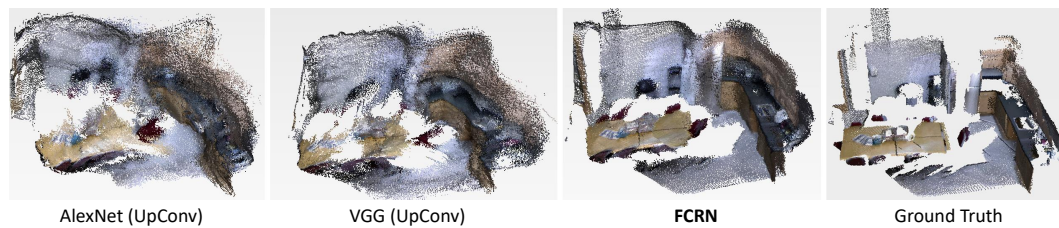


Figure 4.8 3D scene reconstructions from an RGB-(learned) D SLAM framework. In contrast to standard SLAM methods that require an RGB-D camera, we use monocular sequences and predict the corresponding depth map for each frame using our method. The reconstructions are obtained for a kitchen sequence of NYU Depth v2. We compare the reconstructions based on predictions from AlexNet, VGG, FCRN as well as using the Kinect-v2 depth measurements provided by the dataset (ground truth).

We employ a framework inspired by [194, 198]. Camera tracking is carried out via a direct approach with Gauss-Newton optimization, given the photometric differences between successive pairs of frames, as in [198]. Similar to [194], updating the global 3D model of the environment with new depth estimates is carried out in real time via point-based fusion. Both approaches assume RGB-D cameras, which allow to acquire depth measurements, which are then gradually fused into the global model. Our modification is *monocular*, *i.e.* instead of RGB-D sensors, we use the depth predicted by our model, frame-by-frame without additional temporal constraints. To the best of our knowledge, this has been the first time to apply monocular depth estimation to SLAM.

In Figure 4.8 we show the reconstructions that different depth prediction models yield in a kitchen sequence from NYU Depth v2. We compare among AlexNet, VGG and the proposed FCRN architecture. The superior quality of FCRN is not only clear in the single-image predictions of Figure 4.6, but now also in the overall reconstruction of the scene. Thanks to the improved decoder design, FCRN exhibits sharper edges and structures and smoother flat surfaces (such as walls), while AlexNet and VGG reconstructions are significantly affected by the noise present in their depth predictions.

Even though the reconstructions are not yet comparable to the ones using real depth measurements (or even depth computed through geometric SLAM or structure-from-motion approaches that exploit temporal information), we believe that this is an interesting demonstration of depth prediction in such applications that merits further investigation. In fact, when comparing the FCRN reconstruction to the one using ground truth depth acquired by a Kinect-v2 sensor, it is immediately noticeable that the former reconstruction is denser. This is because of sensor limitations that the depth prediction model has learned to overcome, resulting in fully dense depth maps. Comparing to geometric methods, learned depth prediction is also less likely to fail in estimating the depth across low-texture surfaces such as walls—especially since it relies on global image context. On the other hand, geometric approaches can be significantly more accurate in regions with high intensity gradients. Hence, we believe that considering the advantages and shortcomings of these two approaches jointly is a subject of interest. In Chapter 5 we will thus revisit depth estimation



Figure 4.9 Application to synthetic defocus. We predict a depth map from a single image and use it to render a synthetically defocused image at various depths. The result is achieved by applying a bilateral filter based on the distance to the plane set in focus by the user.

in the context of SLAM and 3D reconstruction and exploit the advantages of both geometric and CNN-based approaches in a unified system.

Synthetic Defocus

We also demonstrate the applicability of the predicted depth maps in synthetic defocus.

Barron et al. [27] propose a method to simulate a shallow depth-of-field for images captured with a mobile phone camera and produce high quality results similar to those captured with DSLR cameras and low aperture. Their method produces disparity maps from stereo pairs in a bilateral space (*i.e.* resampled pixel space). The estimated disparity is then used for simulating blur for out-of-focus regions.

While the method relies on cellphones with dual rear cameras, here we use the predicted depth maps from monocular images to directly render the artificially defocused image. We show some examples of synthetic defocus using FCRN predictions in Figure 4.9. The defocused image is rendered by applying a bilateral filter with dynamic strength and size, based on the depth distance to the plane in focus. The focus point can be selected by the user.

4.5.3 Experiments on Semantic Segmentation

Next we evaluate the proposed architecture on the task of semantic segmentation.

Dataset

We use the labeled subset of NYU Depth v2 for our experiments, thus training the model on only 795 samples. We additionally perform data augmentations similar to those described in the previous section, but online. We experiment with 4-class and 40-class segmentation. In 4-class, the scene is segmented into sub-ordinate (structure) classes: *wall/vertical structure*, *ground/floor*, *big structure* (*e.g.* furniture) and *small structure*. The 40-class segmentation is more fine-grained and includes the most common object categories found in indoor scenes, *e.g.* *bed, desk, toilet, window*, etc. [142].

Evaluation Metrics

We evaluate our model following commonly used metrics from previous work [93, 260]. With n_{ij} we denote the number of pixels of ground truth class i that are predicted as class j

Method (4-class)	Data	Pixel Acc.	Mean Acc.	Freq. Jaccard	Avg. Jaccard
Gupta et al. [142]	RGBD	78.0	-	65.0	64.0
Eigen & Fergus [93]	RGBDN	83.2	82.0	-	-
FCRN	RGB	82.3	80.8	70.5	68.3
FCRN	RGBD	83.0	81.5	71.4	69.4
FCRN	RGBDN	83.5	81.4	72.1	70.0

Table 4.4 4-class semantic indoor segmentation (labeled NYU Depth v2). We compare to prior work that use, additionally to the color image (RGB), depth (D) and surface normals (N) as input to the network.

Method (40-class)	Data	Pixel Acc.	Mean Acc.	Freq. Jaccard	Avg. Jaccard
Gupta et al. [142]	RGBD	59.1	28.4	45.6	27.4
Long et al. [260]	RGB	60.0	42.2	43.9	29.2
Long et al. [260]	RGBD	65.4	46.1	49.5	34.0
Eigen & Fergus [93]	RGBDN	65.6	45.1	51.4	34.1
Mousavian et al. [298]	RGB	68.0	51.2	-	38.4
FCRN (320 × 240)	RGB	65.7	47.2	50.6	35.3
FCRN	RGBD	66.9	48.9	52.1	36.5
FCRN	RGBDN	67.2	48.4	51.7	36.2
FCRN (512 × 384)	RGB	68.0	49.4	53.0	37.5
FCRN (640 × 480)	RGB	68.5	49.5	53.7	38.2

Table 4.5 40-class semantic indoor segmentation (labeled NYU Depth v2). The performance of our model is comparable to concurrent work [298]. 40-class is a much more challenging setting, which includes rare of small object categories; we show that the size of the input image makes a significant difference with respect to all metrics.

and with $t_i = \sum_j n_{ij}$ the total number of pixels of class i . Then we compute the following metrics:

- Pixel Accuracy: $\frac{\sum_i n_{ii}}{\sum_i t_i}$
- Mean Accuracy (average over per-class accuracies): $\frac{1}{C} \sum_i \frac{n_{ii}}{t_i}$
- Average Jaccard index: $\frac{1}{C} \sum_i \frac{n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}}$
- Frequency weighted Jaccard index: $\frac{1}{\sum_k t_k} \sum_i \frac{t_i n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}}$

The average Jaccard index is also known as mean intersection over Union (mIoU), which is perhaps the most commonly used metric in multi-class semantic segmentation.

Training Details

We train the semantic segmentation models for up to 200 epochs and choose the epoch with the best validation accuracy for evaluation. We use a batch size of 16 and train with stochas-

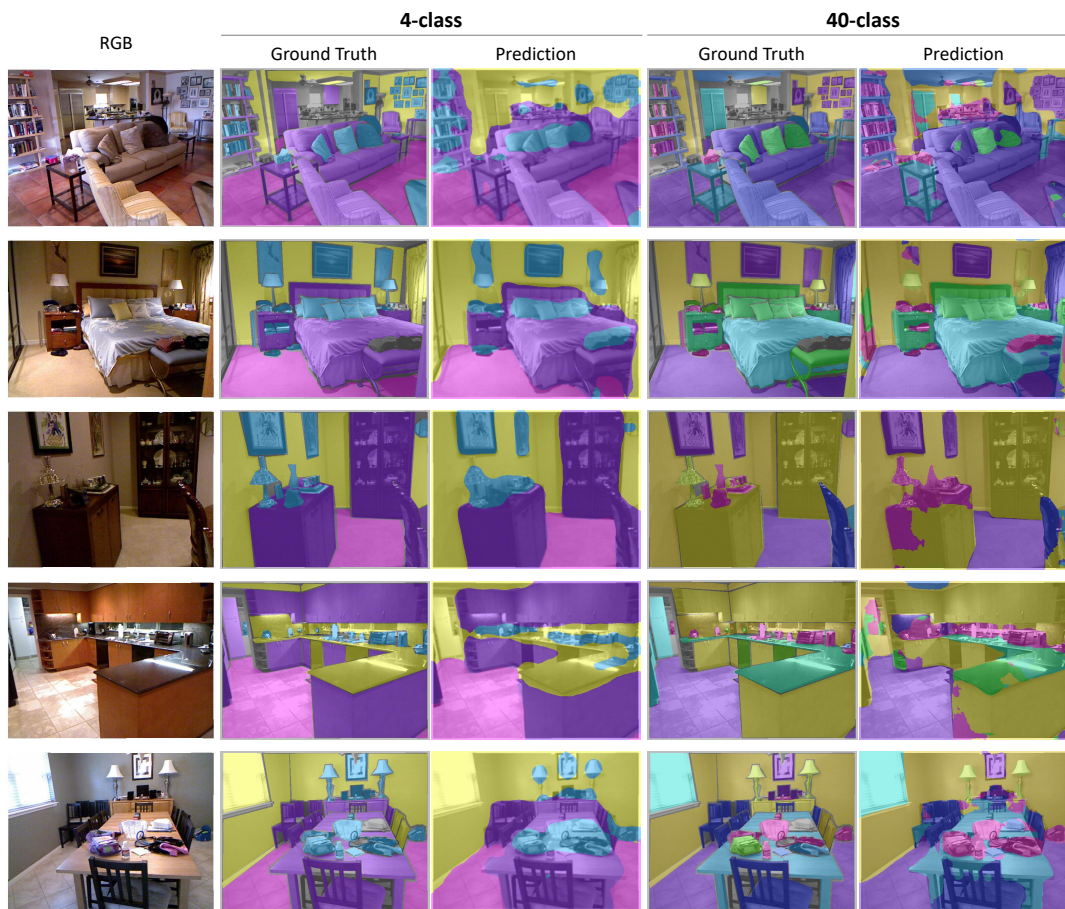


Figure 4.10 Semantic segmentation on NYU Depth v2. We show 4-class and 40-class predictions and the corresponding ground truth labels.

tic gradient descent with momentum equal to 0.9. The initial learning rate is 0.01 which we decrease by half after 50 epochs. Unless otherwise specified, the input resolution is 320×240 pixels.

Besides RGB-only inputs, we also experiment with depth (RGBD) and surface normals (RGBN) as additional information. We concatenate the additional channels to the RGB image, resulting in 4 channels for RGBD and 7 channels in total for RGBN inputs. The normals are extracted from the corresponding depth maps using the toolkit that is available with the dataset. When training these multi-modal variants, we initialize the network with ImageNet pre-trained weights, which consider only RGB inputs and first convolution only has 3 channels. We extract features from the remaining channels in parallel using a separate convolutional layer with 64 filters, which are then added to RGB-extracted features (right after the first layer).

Results

In Table 4.4 we report our experiments on 4-class semantic segmentation. Similar to [93], we also study the effect of different modalities as input to the network. We observe a small

gain in performance when using multi-modal information to train the network. Overall, our method is comparable to [93] for RGBDN inputs.



NOTE: *Multi-modal fusion*

Here we have adopted an *early fusion* scheme for integrating information from different modalities into the network. There are, however, several approaches to this. One way would be to concatenate RGB and/or depth and normal channels to the upsampling layers which potentially has a more direct impact on the output.

In Table 4.5 we also quantitatively evaluate and compare to previous methods on 40-class segmentation. Here, in addition to the multi-modal input, we also experiment with different input sizes. We emphasize that since the proposed architecture is fully convolutional, it does not depend on a fixed input size. Our findings suggest that higher input resolutions (512×384 and 640×480 pixels) improve performance significantly. Our method is comparable to the concurrent work by Mousavian et al. [298] who train with a resolution of 512×512 . It is worth noting that [298] jointly estimates depth and semantic segmentation, and the depth prediction is used to refine the semantic segmentation map in a CRF framework; this significantly improves the performance of the base network.

We demonstrate the segmentation maps predicted by our model for both 4-class and 40-class in Figure 4.10. The model is generally good at distinguishing object categories, even in the more challenging 40-class case. A common source of errors is around object boundaries, which is noticeable by zooming into the figure. The predicted segments do not have sharp edges, especially when it comes to small or detailed structures. This is typically an issue that related work has addressed using skip connections [23, 347] or atrous (dilated) convolutions [55]. Skip connections allow to pass higher-resolution features from the encoding layers to the decoder, thus counteracting the loss of spatial information due to downsampling. On the other hand, dilated convolutions replace downsampling altogether by increasing the field of view of the convolutional with filters with “holes”.

4.6 Conclusion

Building a computer system that can understand a real-world 3D scene from just a single view is a well-established problem in computer vision. There are several aspects that contribute to visual perception, *i.e.* understanding what we see in a scene, which vary from understanding the scene as a whole down to understanding its individual components. Drawing the analogy to visual information processing in the biological brain, these aspects are related to inferring both geometric structures and the semantic meaning and properties of the world around us.

Similarly, in computer vision, two of the most fundamental tasks related to visual perception are depth estimation and semantic segmentation. The goal of the former task is to estimate the distances from the camera to objects in the scene, as well as their structure. Although for

many years this was considered to be an extremely challenging problem, due to its inherent ill-posedness, deep learning methods — given sufficient training data — have led the way to remarkable breakthroughs. One successful approach to this problem is to train deep models on RGB-D data, treating the depth as ground truth and training with full supervision. In this chapter, we contribute to the state of the art on this problem by studying different architectures and comparing fully-connected and encoder-decoder strategies. We propose a fully convolutional residual network, trained with a reverse Huber loss, which, as we show, is better suited for this task than the most common \mathcal{L}_2 objective. Subsequently, we extend the proposed architecture to the semantic counterpart of scene understanding and address the task of semantic segmentation, *i.e.* pixel-wise scene labeling, with a focus on indoor environments.

We have already shown in Section 4.5.2 that it is possible to use the estimated depth map to build a point cloud out of the current view and a preliminary SLAM setup. In the following chapter, we seek a way to improve this by deploying the model in monocular sequences, while building and refining the global representation of the scene incrementally using temporal measurements. If we then also fuse the predicted semantic labels into the global model, it is possible to build a 3D representation of the scene which is semantically meaningful.

Learning in Scene Reconstruction

5.1	Introduction to SLAM	55
5.2	Related Work	57
5.3	Monocular Semantic SLAM	60
5.3.1	Camera Pose Estimation	61
5.3.2	Keyframe Processing	62
5.3.3	Depth Refinement	63
5.3.4	Semantic Reconstruction	64
5.4	Results and Evaluation	65
5.5	Conclusion	68

5.1 Introduction to SLAM

Simultaneous Localization and Mapping (SLAM) is a fundamental problem in computer vision. It aims at estimating a map of an environment (3D scene) and the relative position and orientation of the moving agent (or camera) at the same time. SLAM is therefore a chicken-and-egg problem, as accurate camera pose estimation is needed for mapping but also a map is needed to infer the location of the agent.

Real-time SLAM algorithms have found a wide range of indoor and outdoor applications, in particular for scene reconstruction and navigation of autonomous vehicles (*e.g.* self-driving cars, unmanned aerial vehicles, etc.) or domestic robots (*e.g.* vacuum cleaners), but also in augmented reality and computer graphics. Consequently, SLAM is likely to be a core part of various future intelligent systems.

There exist several SLAM approaches dependent on different data acquisition sensors. The more and the better the sensors that the system is equipped with, the easier the SLAM problem becomes. Multiple cameras and range sensors can be used to directly acquire an accurate RGB-D scan of the current scene. From there on, the main task of a SLAM system is to find correspondences between scans throughout time to integrate them into an ever evolving map. On the other side of the spectrum, SLAM is also possible without range sensors, from a single RGB camera alone. Here one needs to estimate 2D correspondences in the images over time and then use multiple observations of the same group of points together with the theory of multiple view geometry to estimate the 3D configuration of the points and the camera. In this setup one loses the ability to estimate the global scales of the scene and is afflicted by some other limitations, such as the inability to reconstruct when

there is only rotational camera motion. Within the context of this work we focus on visual SLAM from a single RGB camera (no stereo camera setup).

Methods for single RGB cameras such as [76, 302], rely on 2D feature detectors and descriptors. Naturally, feature descriptors and detectors find *key*-points and their correspondences across images. Keypoints are points in the image, that are stable under viewpoint and lighting changes. Such points are usually edges or corners of objects and textures and are thus sparse. This means that the 3D representation that can be reconstructed by those monocular SLAM algorithms will be sparse and mostly defined around corners and edges of objects. Additionally, those methods can only reconstruct the scene up to an unknown scale factor.

Learning-based geometry estimation methods are usually trained on datasets with full 2.5D or 3D supervision — *e.g.* Kinect ground truth data — and have the opposite characteristics. Learning from real data allows the model to estimate the global scale of the scene. Additionally, dense prediction methods are usually good in object centers; but as we have shown in the previous chapter, they are blurry and uncertain around the edges of objects.

This means that both approaches can potentially complement the performance of each other and motivates the approach that we discuss next. Moreover, the ability of modern CNNs to predict semantic information such as object classes and instances can then be used as auxiliary information to enhance the reconstruction and the final map.



KEY CONTRIBUTIONS

We propose an approach to integrate deep depth prediction into keyframe-based direct monocular SLAM that yields significant advantages over geometric-only systems:

- As the depth estimation gives dense outputs, we are able to densely reconstruct the environment, even if it is texture-less.
- Depth estimation (Chapter 4) can provide the 3D reconstruction with an absolute scale, which is now possible to learn from training data, and increase usability in robotics or augmented reality applications.
- Our method is robust to pure rotational motion or slow camera movement, because it does not rely solely on stereo matching (which in turn requires sufficient translation from frame to frame).
- Our geometric reconstruction can be further augmented with semantic labels, which give a *meaning* to the 3D map that goes beyond structure and appearance.

The contents of this chapter have been published in [399].

5.2 Related Work

Visual SLAM has gained research interest since RGB cameras are widespread, compact and cost-effective, thus suitable for practical systems. Typical components of a SLAM system include initialization, tracking and mapping, relocalization, loop closure, pose graph optimization and bundle adjustment. These components are carefully designed and vary in different methodologies.

In a nutshell, *initialization* amounts to defining a global coordinate system in which to perform camera pose and 3D estimation. *Tracking and mapping* are carried out continuously for camera pose estimation. For tracking, correspondences are computed between a frame and the already reconstructed 3D map of the environment and the current camera pose is computed through projective geometry. Then the map is updated with the 3D estimation of newly observed regions. *Relocalization* is important in case of temporary tracking failures which might be, for example, the consequence of fast camera motion. *Loop closure* is used to encourage consistency in the map by detecting when a region has been revisited and computing the accumulated camera trajectory error in the detected loop. Usually, a graph representation for camera poses is preferred and *pose graph optimization* [217] is used to find a globally optimal configuration of poses and reduce errors.

Also related to SLAM are visual odometry and structure-from-motion (SfM). The difference between visual SLAM and visual odometry is that the latter refers to methods for estimating the pose of a moving camera that do not, however, optimize for global consistency in the map. As we have discussed in the previous chapter, SfM usually refers to methods that reconstruct a model from a collection of images taken at various viewpoints.

When it comes to the input modality, SLAM systems exist that rely only a single camera/view — *i.e.* monocular — as opposed to stereo cameras or RGB-D sensors. Comparing to other sensors, such as laser scanners, conventional cameras also only have a limited field of view of the environment, which makes the problem more challenging. Another challenge is that a single view does not properly capture scene geometry, making accurate reconstruction difficult due to scale drifts over time. At the same time, this very challenge constitutes a significant advantage of monocular SLAM: it is scale-independent, thus the same algorithm can operate on both indoor and outdoor environments and can even transition between them.

Since there are no direct depth measurements in the monocular case, temporal information can be exploited to estimate *relative* depth. As the camera moves over time, consecutive views are considered as a stereo pair and depth can be estimated via small-baseline stereo matching. To find the real scale of the environment, some approaches use 3D object detection given objects of known sizes [113]. In our case, we can inject an absolute scale into the reconstruction a data-driven approach that learns the typical geometry of specific types of environments, *e.g.* indoor spaces.

Depending on how the input frames are processed, monocular SLAM algorithms can be distinguished as feature-based or direct (feature-less).

Feature-based Approaches

Feature-based methods operate on distinct image key-points, extracted by descriptors such as SIFT [261] or SURF [33], to estimate camera pose and geometry. The use of invariant features — *e.g.* invariance to scaling, illumination or viewpoint — should allow for accurate matching. On the other hand, a lot of information about the scene, that is not compliant with the chosen feature characteristics, is discarded and the resulting depth maps are usually very sparse. Common in feature-based methods is also *bundle adjustment* [411] which is used to jointly optimize the geometry and camera poses by minimizing the map reprojection error.

Originally, monocular SLAM was addressed by filtering, for example, extended Kalman filters (EKF). The map is represented by a state vector consisting of the camera pose and 3D locations of feature points and a co-variance matrix. The first approach was MonoSLAM [76]. The drawback of this method is the amount of computation required which is proportional to the size of the environment.

One of the most representative feature- and keyframe-based methods is PTAM [207]. From a computational perspective, PTAM proposes to separate tracking and mapping into parallel threads. Decoupling mapping and tracking means that not all frames need to be used to increment the map, but only a conservative number of selected *keyframes*, which are refined via bundle adjustment. The original method was limited to small-scale environments.

ORB-SLAM [303] is one of the state-of-the-art approaches of this category. It uses ORB features [351] which are fast to compute and match and invariant to viewpoint changes; these are used in all modules, including loop closure and camera relocalization. Invariance to viewpoint further improves bundle adjustment because it allows for large-baseline matching. ORB-SLAM2 extends these ideas to RGB-D and stereo [302].

Direct Approaches

Direct methods (feature-less) perform camera pose and geometry estimation by directly optimizing over image intensities and are able to create semi-dense 3D maps of the environment. Such methods do not abstract images into features, but leverage all information available in the image; therefore, they are typically more robust in low-textured environments, but computationally more expensive. Furthermore, since their reconstructions are more detailed they can be useful in various applications, for example in augmented reality. One downside is that direct approaches assume mostly static scenes and rely on photometric consistency, thus they can be sensitive to illumination changes.

One instance of these methods is DTAM [307], which computes fully dense depth maps in real time on a GPU. Tracking in DTAM is formulated as a registration problem between a frame and views from the dense 3D model. Depth estimation is carried out from a set of frames through multi-baseline stereo, followed by total variation regularization so that the predicted depth is smooth in uniform intensity regions.

On the other hand, LSD-SLAM [97], which is based on the visual odometry work of Engel et al. [98], proposes a probabilistic semi-dense map representation. Geometry is estimated from high-intensity gradients only, which reduces complexity and runs in real time on a CPU. The multi-level mapping algorithm [131] builds on LSD-SLAM improving its density while retaining comparable efficiency.

Learning-based Approaches

While prior to the publication of this work [399] there had been little progress in combining machine learning with SLAM, it is now an emerging field and several methods have been proposed that replace hand-designed components of traditional SLAM workflow with learnable models. The advantage of deep learning models comes from the fact that they can discover powerful representations for solving the task at hand. Consequently, apart from monocular depth estimation, deep networks have been trained to predict camera motion, optical flow and even end-to-end visual odometry. They can be also used to augment a purely geometric reconstruction with semantic entities. Davison [75] refers to such hybrid technology as “Spatial AI” as it allows embodied agents to build meaningful scene representations for smartly navigating and interacting with their environment in real time.

Deep Visual Odometry and Depth Estimation One family of methods focuses on deep visual odometry, *i.e.* replacing the conventional pose estimation pipelines, that heavily rely on feature extraction and matching, with deep networks. DeepVO [433] and VidLoc [68] estimate camera poses from a video stream in an end-to-end manner using CNN-RNN architectures that are trained with pose supervision. In general, the CNN is used for extracting per-frame features, while the RNN is responsible for learning temporal dependencies. DeMoN [415] additionally predicts and iteratively refines depth maps with a single architecture that is trained from stereo sequences. DeepTAM [485] conceptually follows its classic counterpart DTAM [307], though the tracking and mapping modules are implemented with neural networks. Comparing to other methods, it does not only perform visual odometry but also has a strong mapping component that refines the cost volume using multiple frames.

To reduce the amount of annotations that are required, many authors have focused on the self-supervised setup, *i.e.* using image reconstruction as supervision instead of direct regression. The focus of this line of work is the joint estimation of camera motion and depth — also discussed in Section 4.2. Zhou et al. [488] propose one of the first approaches to learn camera motion and depth simultaneously from just a video stream through differentiable warping of RGB inputs based on the idea of spatial transformers [180]. Their method relies on the assumption of small-baseline stereo between consecutive frames and thus can only estimate relative distances. Following a similar approach, Li et al. [244] show that it is possible to use stereo pairs at training time (while being monocular at test time) to yield predictions with absolute scale instead. As such methods rely on photometric consistency as an objective function — which might not hold true in practice — Zhan et al. [477] propose, in addition, a feature reconstruction loss. However, the camera pose estimation of these methods is still not as accurate as that of geometric approaches. For this reason, Yang et al. [464] propose a semi-supervised method and exploit geometric methods to generate data for partial super-

vision. Wang et al. [427] propose a module for differentiable visual odometry that can be used within a network, initialized either by a predicted or the identity pose; their goal is to decouple self-supervised depth estimation from ego-motion.

Learning Representations Focusing on dense reconstructions, CodeSLAM [41] proposes to represent scene geometry with a compact, deep code that is optimizable; the representation is constructed by a variational auto-encoder (VAE) [203] on depth images that is further conditioned on image features. The codes can be optimized during bundle adjustment to achieve global consistency. Focusing on the mapping component, MapNet [155] introduces a representation for 3D environments that is fully integrated with deep learning techniques. In this work, an RNN performs map updates from an RGB-D inputs while optimizing localization accuracy. Localization and registration are intuitively represented by dual operators: a convolution and a deconvolution, respectively. Gupta et al. [141] also propose a deep network to encode the environment into a latent spatial memory and further focus on navigation with a differentiable planner.

Last but not least, related to our work is SemanticFusion [286], which also addresses the fusion of CNN-predicted semantic labels into the geometric model.

5.3 Monocular Semantic SLAM

In the following, we explore how to integrate CNN predictions (*e.g.* from Chapter 4: geometry and semantics) into monocular SLAM to achieve semantic 3D reconstruction of a scene.

An overview of the method is shown in Figure 5.1. Following *keyframe-based* SLAM frameworks [97, 207, 303], we select only a few frames which contribute to the pose graph optimization and from which we also infer dense depths maps using the previously discussed learned approach. New keyframes are selected when the estimated camera pose is sufficiently different from those of past keyframes. Estimating the camera pose is described in detail in Section 5.3.1. To integrate a predicted depth map into the previously reconstructed geometry, a confidence map is computed relative to the depth map of the nearest keyframe (Section 5.3.2). However, as we have already seen, depth estimation tends to produce blurry results — especially around object boundaries — and particularly noisy planar surfaces, such as walls, that yield sub-par reconstructions (Figure 4.8). Therefore, a crucial stage of our approach deals with how to recover details in the predicted keyframe depth maps (Section 5.3.3). Finally, semantic labels can be also fused into the global model, resulting in an incrementally built semantic scene reconstruction (Section 5.3.4).

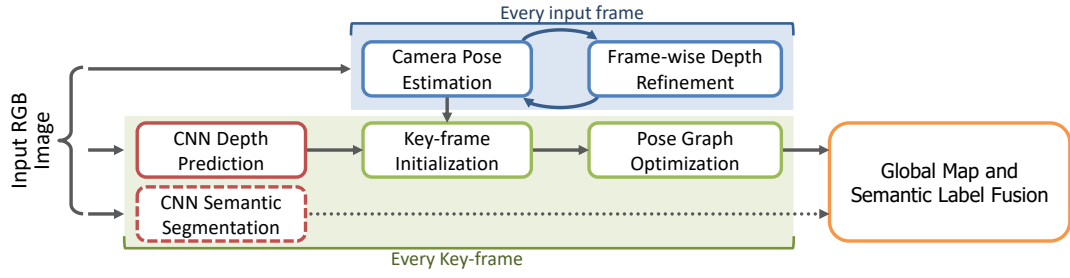


Figure 5.1 Overview of CNN-SLAM. A CNN predicts depth and semantics for keyframes, which are then fused into a geometric SLAM framework for camera pose estimation and frame-wise refinement.

5.3.1 Camera Pose Estimation

To estimate the position and orientation of a moving camera, we closely follow the paradigm of LSD-SLAM [97], where the structural elements are keyframes $k_1, \dots, k_n \in \mathcal{X}$ and their respective poses \mathbf{T}_{k_i} and depth maps D_{k_i} .

For every frame t , we want to estimate its relative camera pose with respect to the *nearest* keyframe k_i , $\mathbf{T}_t^{k_i} = [\mathbf{R}_t, \mathbf{t}_t] \in \mathbb{SE}(3)$, where $\mathbf{R}_t \in \mathbb{SO}(3)$ is the rotation matrix and $\mathbf{t}_t \in \mathbb{R}^3$ is the translation vector. When the relative pose is known, the corresponding camera pose in the world coordinate system can be computed as $\mathbf{T}_t = \mathbf{T}_t^{k_i} \mathbf{T}_{k_i}$.

We denote an image point as $\mathbf{p} = (x, y) \in \Omega \subset \mathbb{R}^2$ with a corresponding depth value $d = D(\mathbf{p})$. $\tilde{\mathbf{p}}$ indicates homogeneous coordinates. Let \mathbf{K} be the intrinsic camera matrix, then the position of a point in 3D can be computed from the keyframe depth map D_{k_i} as

$$\mathbf{V}_{k_i}(\mathbf{p}) = \mathbf{K}^{-1} \tilde{\mathbf{p}} D_{k_i}(\mathbf{p}). \quad (5.1)$$

Furthermore, a projection function is used to map a 3D point to its 2D coordinates in image space:

$$\pi((x, y, d)^T) = (x/d, y/d)^T. \quad (5.2)$$

Then, we can compute the transformation between the two images I_t and I_{k_i} by minimizing the photometric error

$$r(\mathbf{p}, \mathbf{T}_t^{k_i}) = I_{k_i}(\mathbf{p}) - I_t(\pi(\mathbf{K} \mathbf{T}_t^{k_i} \mathbf{V}_{k_i}(\mathbf{p}))), \quad (5.3)$$

that is

$$\min_{\mathbf{T}_t^{k_i}} \sum_{\tilde{\mathbf{p}} \in \Omega} \left\| \frac{r(\tilde{\mathbf{p}}, \mathbf{T}_t^{k_i})}{\sigma(r(\tilde{\mathbf{p}}, \mathbf{T}_t^{k_i}))} \right\|_{\delta}, \quad (5.4)$$

where $\|\cdot\|_{\delta}$ denotes the Huber norm and $\sigma(\cdot)$ computes the residual's variance as in [97]. Equation (5.4) gives the maximum likelihood estimate for $\mathbf{T}_t^{k_i}$ following Gauss-Newton optimization. In practice, since our depth maps are dense, we only use a subset of points $\tilde{\mathbf{p}} \subset \mathbf{p} \in \Omega$ when computing photometric errors; these are the pixels in high intensity gradient regions.

5.3.2 Keyframe Processing

Depth Initialization If the estimated camera pose for the current frame is too far from that of the nearest keyframe (as defined by a threshold [97]), a new keyframe is created and it is then used for computing the relative pose of subsequent frames. For a new keyframe, a dense depth map can be estimated using the single-image approach described in Chapter 4, *i.e.* $D'_{k_i} = \text{CNN}(I_{k_i})$. Temporal information is not taken into account.

As the sensor used during reconstruction might differ from the one used to acquire the data that the CNN was trained with, we rescale the predicted depth map as

$$D_{k_i}^{\text{init}} = \frac{f_{\text{cur}}}{f_{\text{pre}}} D'_{k_i}, \quad (5.5)$$

where f_{cur} is the focal length of the current camera and f_{pre} is the focal length used for training the network. The importance of this adjustment can be observed in Figure 5.2 (a) and (b).

Uncertainty Estimation Each keyframe k_i , with its depth map D_{k_i} , is further linked to a heatmap U_{k_i} that acts as an uncertainty proxy. We note that, here, we use the term *uncertainty* not with a probabilistic meaning, but rather in the sense of reliability of the predicted depth map. The initial value of U_{k_i} is set to the difference between the depth value of points in the current keyframe k_i and the corresponding points in the nearest keyframe k_j . The difference can be computed after warping D_{k_j} to be aligned with D_{k_i} using the estimated relative pose $T_{k_j}^{k_i}$. Let \mathbf{p}' be the warped coordinates from k_j to k_i , *i.e.*

$$\mathbf{p}' = \pi \left(\mathbf{K} T_{k_j}^{k_i} \mathbf{V}_{k_j}(\mathbf{p}) \right), \quad (5.6)$$

then the uncertainty map for keyframe k_i can be defined as

$$U_{k_i}^{\text{init}}(\mathbf{p}) = \left(D_{k_i}^{\text{init}}(\mathbf{p}) - D_{k_j}(\mathbf{p}') \right)^2. \quad (5.7)$$

By definition, this map holds information about how much the predicted depth varies across slightly different views of a scene. A high value means that the predicted distance for the same point has changed considerably between the two views—a phenomenon that, for example, often occurs on wall surfaces and other low-texture regions. This implies that D_{k_i} is less reliable in these regions. Low values indicate points where the depth is similar in the two frames.



NOTE: *Uncertainty Estimation*

There are other ways to compute uncertainty for depth predictions—and generally CNN models. One example is the Bayesian approach of [196] that models epistemic and aleatoric uncertainties. However, this approach is too slow for real-time applications, because it requires on several forward passes for a single data point. Another possibility is the multiple hypothesis framework of Rupprecht et al. [357], which we

discuss in Chapter 6. In this case, prediction uncertainty can be approximated by computing a variance map across multiple predictions for the same input.

Uncertainty Propagation and Depth Fusion The accuracy of a new keyframe depth map can be improved using depth information propagated from the nearest keyframe $D_{k_j}(\mathbf{p}')$, as the latter will have already been subject to refinement at this point (details in Section 5.3.3).

We also compute the propagated uncertainty map from the nearest keyframe k_j :

$$\tilde{U}_{k_j}(\mathbf{p}') = \frac{D_{k_j}(\mathbf{p}')}{D_{k_i}^{\text{init}}(\mathbf{p})} U_{k_j}(\mathbf{p}') + \sigma_p^2, \quad (5.8)$$

where σ_p^2 is white noise variance [98]. Finally, the two depth maps are merged weighted by the computed uncertainty. In regions where the current depth prediction is unreliable (high uncertainty), we leverage depth propagated from the nearest keyframe instead, weighting $D_{k_j}(\mathbf{p}')$ by $U_{k_i}(\mathbf{p})$, and vice versa. Therefore, the fusion scheme that yields the final depth map and its corresponding uncertainty is:

$$D_{k_i}(\mathbf{p}) = \frac{\tilde{U}_{k_j}(\mathbf{p}') D_{k_i}^{\text{init}}(\mathbf{p}) + U_{k_i}^{\text{init}}(\mathbf{p}) D_{k_j}(\mathbf{p}')}{U_{k_i}^{\text{init}}(\mathbf{p}) + \tilde{U}_{k_j}(\mathbf{p}')} \quad (5.9)$$

$$U_{k_i}(\mathbf{p}) = \frac{U_{k_i}^{\text{init}}(\mathbf{p}) \tilde{U}_{k_j}(\mathbf{p}')}{U_{k_i}^{\text{init}}(\mathbf{p}) + \tilde{U}_{k_j}(\mathbf{p}')} \quad (5.10)$$

Pose Graph Update When a new keyframe is created, a new node is also added to the pose graph, with edges connecting to existing keyframes with a similar viewpoint. The pose graph is continuously refined through pose graph optimization [217].

5.3.3 Depth Refinement

We have seen that the scene geometry is constructed by a set of depth maps which are predicted for each keyframe. Frames that are not selected as keyframes contribute to the refinement of the depth map of the current keyframe.

We follow the direct, semi-dense approach of Engel et al. [98] to estimate depth from video using small-baseline stereo comparisons. This approach produces accurate, probabilistic depth estimates (D_t, U_t) for image regions with high intensity gradients (semi-dense) and thus can be seen as complementary to CNN predictions that are fully dense but suffer with finer details around object boundaries.

For every frame t , D_t and U_t are continuously used to refine the keyframe through a similar fusion scheme as before, this time between D_{k_i}, U_{k_i} and the current frame after it has been warped with estimated pose $\mathbf{T}_t^{k_i}$.

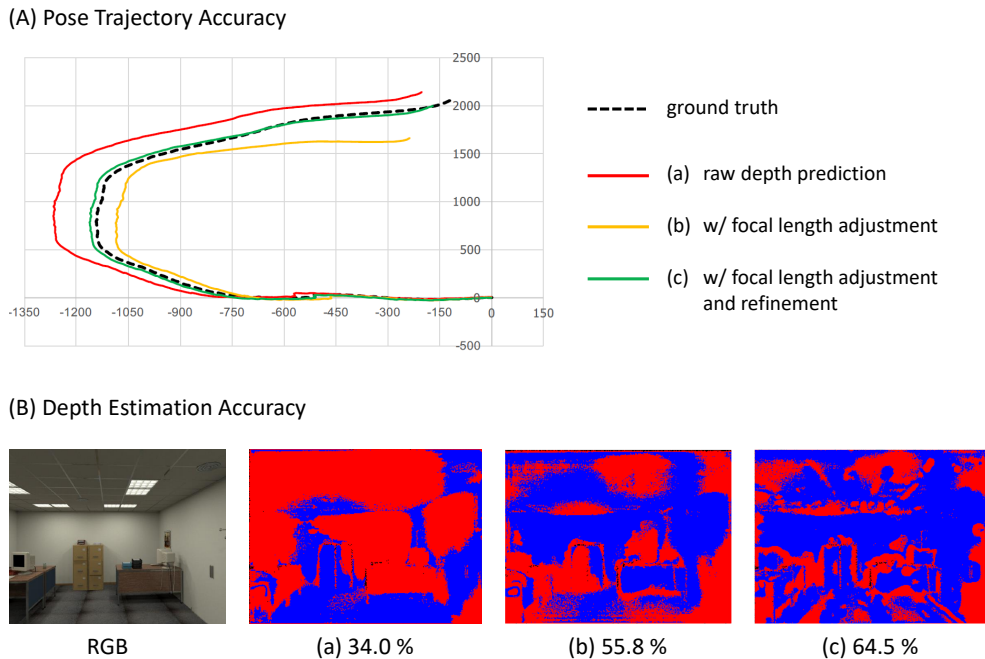


Figure 5.2 Component evaluation. Comparison of reconstruction performance using (a) raw depth predictions, (b) focal length adjustment (Equation (5.5)) and (c) adjustment and frame-wise depth refinement (Section 5.3.3), in terms of (A) pose trajectory accuracy and (B) depth estimation accuracy. In the binary maps (B), blue indicates correctly estimated depth (within 10 % of the ground truth depth values) and red indicates wrong predictions. The comparison is shown on a sample sequence of the ICL-NUIM dataset [147].

 **NOTE:** *Pure Rotational Motion*

Triangulation, *i.e.* estimating a point in 3D space from image point correspondences, requires a sufficient baseline between two camera views. In case of pure rotations (no translation), or even very slow camera motion, a baseline between the two views cannot be established, thus (inverse) depth cannot be estimated. Direct monocular SLAM systems such as [97, 98] would fail, as they solely depend on stereo matching for estimating depth. On the other hand, a data-driven approach overcomes this failure; in case of rotational-only motion the keyframe will simply retain the depth map estimated by the CNN and will not be affected by unreliable stereo-matching estimates.

The effectiveness of this refinement on a sample sequence from the ICL-NUIM dataset is demonstrated in Figure 5.2 (c).

5.3.4 Semantic Reconstruction

Apart from bringing an absolute scale to the reconstruction, one of the major advantages of deep learning-augmented SLAM is the ability to enhance the estimated map of the environment with semantic meaning via object recognition and segmentation. This enables agents

to not just navigate in an environment, but actually gain knowledge about individual objects and their constellations and potentially also how to interact with them.

When semantic labels are available (for example after semantic segmentation of the scene from single images), they can be incorporated into the 3D model of the scene, *i.e.* densely labeling the geometric reconstruction. To achieve this, we modify the method proposed by Tateno et al. [401] to incrementally fuse depth and semantic segmentation maps. Temporal averaging of label assignments is performed for each 3D point to reduce noise occurring from frame to frame and the most probable (most frequent) label is chosen for each point. Hence, the resulting semantic labeling of the 3D model will generally be more accurate than the individual CNN predictions. In [401] object segments are geometrically computed from surface normal edges. Here, we use our proposed CNN architecture (Section 4.4) to predict *semantic* segmentation maps frame by frame. While our model is trained with semantic labels annotated for a limited amount of images on NYU-Depth-v2 [306], it generalizes well even to out-of-domain data such as the synthetic scenes of ICL-NUIM [147] used in our SLAM experiments. We use the 4-class segmentation model described in Section 4.5.3 to segment scenes into the supercategories (*floor, vertical structure, large structure, small structure*).

5.4 Results and Evaluation

To evaluate our method, we carry out experiments on two benchmark datasets, ICL-NUIM [147] (synthetic) and TUM-RGBD [387] (real). The datasets provide camera trajectories and depth maps as ground truth, thus we are able to quantitatively evaluate our approach in terms of tracking and reconstruction accuracy.

Implementation Details We use two separate CNNs for depth estimation and semantic segmentation, both trained on NYU-Depth-v2 data [306] as discussed in the previous chapter. The predicted maps are resized to a resolution of 320×240 pixels. Although the CNNs are trained on a different dataset—some major differences being the acquisition sensor and scene layouts—they show good generalization capability. Besides that, the predictions are further improved within the SLAM framework (through frame-wise refinement for depth estimation and temporal smoothing for the semantic segmentation).

Since the CNN forward passes are only required for new keyframes, the method is efficient and runs in real-time with the CNN predictions carried out in parallel on a GPU and all SLAM stages on CPU threads.

Comparison with State-of-the-art SLAM

In Table 5.1 we report quantitative comparisons among various methods—direct LSD-SLAM¹ [97], feature-based ORB-SLAM² [303] and REMODE³ [321])—on selected sequences

¹https://github.com/tum-vision/lsd_slam

²https://github.com/raulmur/ORB_SLAM2

³https://github.com/uzh-rpg/rpg_open_remode

Method	ICL-NUIM							TUM-RGBD			
	office0	office1	office2	living0	living1	living2	mean	seq1	seq2	seq3	mean
Absolute Trajectory Error (lower is better)											
LSD [97]	0.528	0.768	0.794	0.516	0.480	0.667	0.626	1.826	0.436	0.937	1.066
LSD-BS [97]	0.587	0.790	0.172	0.894	0.540	0.211	0.532	1.717	0.106	0.037	<u>0.620</u>
ORB [303]	0.430	0.780	0.860	0.493	0.129	0.663	0.559	1.206	0.495	0.733	0.811
FCRN [227]	<u>0.337</u>	<u>0.218</u>	0.509	<u>0.230</u>	<u>0.060</u>	0.380	<u>0.289</u>	<u>0.809</u>	1.337	0.724	0.957
CNN-SLAM [399]	0.266	0.157	<u>0.213</u>	0.196	0.059	<u>0.323</u>	0.202	0.542	<u>0.243</u>	<u>0.214</u>	0.333
% Accurate Depth Pixels (higher is better)											
LSD-BS [97]	0.603	4.759	1.435	1.443	3.030	1.807	2.180	3.797	3.966	6.449	4.737
REMODE [321]	4.479	3.132	16.708	4.479	2.427	8.681	6.651	9.548	12.651	6.739	9.646
FCRN [227]	<u>17.194</u>	<u>20.838</u>	<u>30.639</u>	15.008	<u>11.449</u>	33.010	<u>21.356</u>	12.982	<u>15.412</u>	<u>9.450</u>	<u>12.615</u>
CNN-SLAM [399]	19.410	29.150	37.226	<u>12.840</u>	13.038	<u>26.560</u>	23.037	<u>12.477</u>	24.077	27.396	21.317

Table 5.1 Comparison with state-of-the-art SLAM methods. We compare our full approach (CNN-SLAM) and baseline (FCRN) with previous methods on two datasets (ICL-NUIM and TUM-RGBD) under two metrics: absolute trajectory error (in meters) and the percentage of accurately estimated depth pixels. (TUM-RGBD seq1: fr3/long_office_household, seq2: fr3/nostructure_texture_near_withloop and seq3: fr3/structure_texture_far.

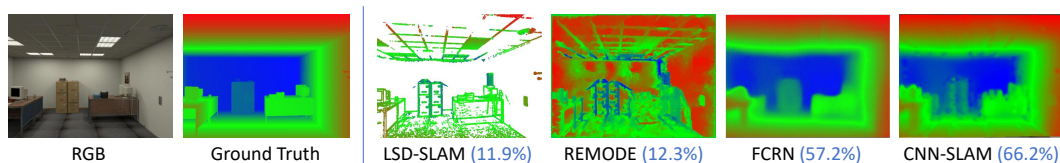


Figure 5.3 Comparing depth map density and accuracy on the office2 sequence of ICL-NUIM. FCRN shows the raw depth prediction for the selected frame, while CNN-SLAM is the result after keyframe refinement. The reported % are the correctly estimated depth pixels on this frame.

of the two benchmark datasets. Since monocular SLAM methods only estimate depth up to scale, we initialize the scale for [97] using the available ground truth depth maps (LSD-BS). REMODE is a method focusing on dense reconstructions and requires the camera pose as input, which we provide using LSD-BS.

Together with our full system (CNN-SLAM), we also evaluate the performance of a baseline (FCRN) which simply uses the depth predictions as input to an RGB-D SLAM system [194] without any refinement (qualitative results of this baseline are shown in Figure 4.8).

We evaluate all methods under two metrics; the absolute trajectory error is computed as the root mean square error between predicted and ground truth camera translations, while the percentage of accurate depth pixels refers to amount of depth values that fall within 10% of the ground truth depth and is used to evaluate both the density and accuracy of the reconstruction.

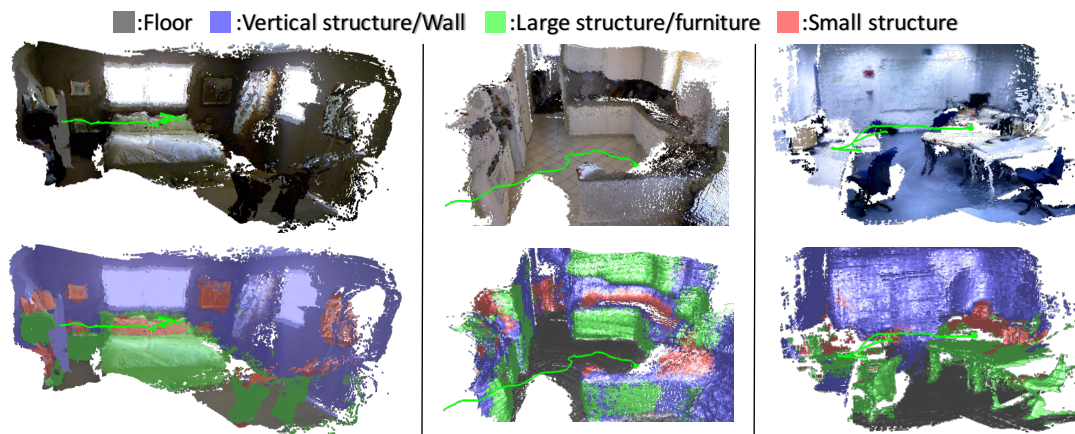


Figure 5.4 Geometric and semantic reconstructions. We show the geometric and semantic reconstructions of our method on three scenes: NYU-Depth-v2 bedroom_0115 (left), kitchen_0046 (middle) and an office scene from a custom setup (right). Camera trajectory is shown in green.



Figure 5.5 Reconstruction under pure rotational motion. We compare LSD-SLAM and CNN-SLAM reconstructions of a sequence (TUM-RGBD fr1/rpy) that consists of rotational motion.

Our approach of combining a direct SLAM system with a deep network for depth prediction performs best under both metrics. In particular, the absolute trajectory error is even lower than LSD-BS, even though ground truth information is used in the latter for bootstrapping. As for the reconstruction accuracy, CNN-SLAM is significantly more powerful than LSD-BS and REMODE as it is more dense, but also outperforms FCRN-based SLAM, suggesting the effectiveness of the depth refinement scheme. A qualitative comparison in terms of depth density and quality is also shown in Figure 5.3.

Qualitative Results

Qualitatively, 3D semantic scene reconstructions from our system are shown in Figure 5.4. The left and middle scenes are sequences from the test set of NYU-Depth-v2 and the scene on the right is acquired using an own setup. The predicted camera trajectory is shown in green.

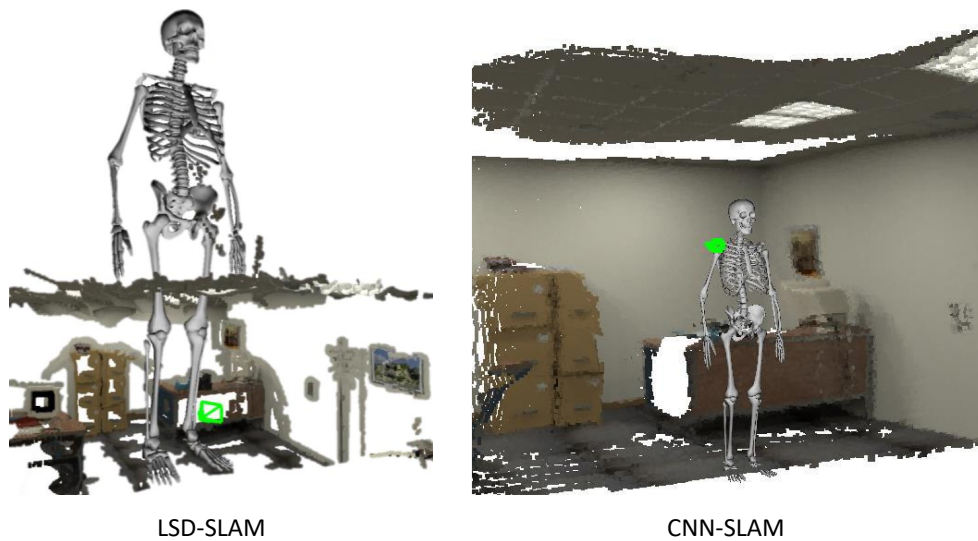


Figure 5.6 Reconstruction scale. We augment an object (skeleton) in the reconstructed environments of LSD-SLAM and CNN-SLAM to show the important of real-scale reconstructions in applications such as augmented reality.

Pure Rotational Motion

To demonstrate the effectiveness of our approach under pure rotations, we show a qualitative example in Figure 5.5. We compare the reconstructions of LSD-SLAM and CNN-SLAM on the *fr1/rpy* sequence of TUM-RGBD which mainly consists of rotational motion. Under this challenging scenario for monocular SLAM systems, our method remains robust thanks to the CNN predictions that are unaffected by any kind of motion. However, the overall quality of the reconstruction is lower as the blurry CNN predictions cannot get sufficiently refined. Under pure rotational motion, the performance of the system becomes comparable to the preliminary SLAM experiment that was discussed in Section 4.5.

Absolute Scale

Finally, in Figure 5.6 we qualitatively show that absolute-scale reconstructions are preferred in applications such as augmented reality. Our method achieves a real-scale monocular reconstruction that is also very accurate, thus objects of known sizes can easily be augmented in the scene, in contrast to geometric monocular SLAM approaches. One limitation of the use of learning in SLAM is that, since absolute scale is inferred from training examples, the network must be trained with scenes from the same type of environment (*e.g.* indoors).

5.5 Conclusion

Total scene understanding is central to applications such as navigation and localization in both indoor and outdoor environments. In Chapter 4 we discussed a deep learning method to address two fundamental tasks in scene understanding; depth estimation and semantic segmentation. Crucially, this enables the recovery of geometric and semantic scene proper-

ties from just a single image. In this chapter, we have demonstrated the advantage of this approach in reconstructing the 3D model of the scene from a monocular sequence. In particular, we address this problem by considering two different perspectives and leverage their complementary nature.

On one hand, **traditional monocular SLAM systems** rely on point correspondences or image similarities between frames to estimate relative camera motion. Then depth information can be estimated through projective geometry. Some challenges that arise in these methods are low depth density (especially across texture-less image regions), failure under pure rotational motion and absence of a metric scale.

On the other hand, **deep learning models** trained for monocular depth estimation can yield dense depth maps from single images without relying on specific camera motion. Moreover, it is possible to estimate an absolute scale for the scene, which the model learns from experience—provided many RGB-D examples from a specific type of environment (*e.g.* indoors). Although significant progress has been made in the last few years to improve the structural quality of depth maps, the disadvantage of learning approaches is that the predictions are usually not sharp enough for scene reconstruction, which is further amplified when the models are deployed in out-of-domain environments (*e.g.* scenes that are not similar to the ones used during training).

The framework that we have presented in this chapter takes advantage of the complementary nature of the two approaches to tackle their respective shortcomings. We leverage learned depth prediction on keyframes, which results in dense maps with absolute scale and is robust to rotational motion and scenes with low texture. Taking advantage of the temporal component, the predicted depth maps are continuously refined to recover depth details via small-baseline stereo matching which is especially strong along image gradients. We show that this combination yields dense and accurate 3D models of indoor environments, which is a direct application and an example of the usefulness of depth estimation in higher-level tasks. Last but not least, integrating the *semantic* counterpart of our approach into the geometric framework comes with the significant advantage that we can now give *meaning* to the reconstruction. This is vital in applications that involve intelligent agents navigating and acting within the scene.

Object-centric Understanding

6.1	Motivation	71
6.2	Localization of Surgical Instruments	73
6.2.1	Methodology	73
6.2.2	Results and Evaluation	76
6.3	Localization of Grasping Points	81
6.3.1	Methodology	82
6.3.2	Results and Evaluation	86
6.4	Conclusion	92

6.1 Motivation

So far we have presented solutions to computer vision problems that collectively help to equip intelligent systems with a global understanding of real-world scenes. In the context of this dissertation, a scene is considered to be a view of a (usually complex) environment in the real world, which consists of several objects that interact with each other.

However, in real-world applications, it is often the case that the system might be tasked with an object-centric problem. Perceptual tasks at object level might be different than those at scene level. For example, when facing an entire scene, a human as well as a computer might want to recognize the environment or how to navigate in it, based on the geometry, affordances or other semantic properties of the objects that compose the scene as a whole. On the other hand, when addressing problems at object level, the field-of-view is usually more narrow and the focus is on specific objects, rather than entire scenes, as well as how these objects interact with the scene or how an external agent might act upon them. Visual perception at object level can be useful in several applied scenarios, providing more in-depth understanding of their properties.

One such example is industrial robots. Although still dominated by rigid hand-engineered programs, machine learning is rapidly revolutionizing industrial robotics, through object recognition and imitation learning, helping teach robots manipulation tasks such as picking and placing objects. The setting under which such robots operate is usually restricted to a pre-specified workplace, which is very different to the busy, dynamic environment of natural scenes. In this case, it is more relevant to recognize individual objects and understand their physical properties.

Another example is computer-aided medical interventions, where successful recognition and tracking of surgical instruments is critical. In this case, although the overall “scene” is

actually a very dynamic environment, the effective field of view is typically limited. The object in focus is the surgical instrument(s) in view, while the scene is the underlying anatomy that the instrument acts upon. The use of computer vision and machine learning in robotic-assisted surgical systems allows for great control and assistance to the surgeon without additional equipment, such as markers. For example, segmentation of surgical instruments can be used to understand the region of interest for potential augmented reality overlays (or displaying critical information) without occluding the surgeon's view. Tracking of surgical instruments can also improve surgical workflow analysis or provide information for vision-based surgical robot control.

In this chapter, we discuss two applied scenarios of object-centric scene understanding. In Section 6.2 we take a closer look into the medical scenario and propose a method for simultaneous segmentation of the surgical instrument and localization of its articulation mechanism — for example joints of the end-effector. This method has been published in [226].

In Section 6.3 we delve into the application of robotic grasping. The ability to grasp and manipulate objects is necessary in the field of personal robotics but also, as mentioned above, in industrial manufacturing. Here we focus on vision-based grasp detection, *i.e.* accurately localizing possible grasping points directly from image data, without the use of tactile or depth sensors. Our method has been published in [118].



KEY CONTRIBUTIONS

In the context of the aforementioned object-centric applications, we present two approaches to *localization*, taking into account the specific characteristics of each problem. Our contributions are the following:

- We cast localization as a spatial, high-dimensional problem, by representing points of interest as heatmaps. We show that this formulation allows to better exploit global image context and explicitly models the spatial uncertainty that arises from ambiguous labeling.
- In surgical instrument tracking, we address the task of simultaneous segmentation and localization by adapting the previously proposed architecture (FCRN) to this multi-task setting. We also thoroughly evaluate different variants to justify the choices that lead to the heatmap-based formulation.
- In robotic grasping, we take an additional step to model the ambiguity of the task. Since many grasp configurations might be plausible, we extend our model to predict multiple grasping hypotheses. We show that this formulation leads to more effective training of the model *and* gives the robot several possibilities at test time.

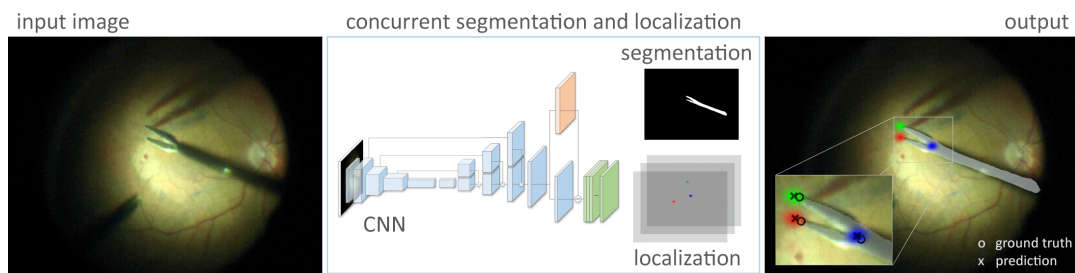


Figure 6.1 Overview of the proposed method. We jointly predict a segmentation map and the landmark positions of a surgical tool by adapting FCRN.

6.2 Localization of Surgical Instruments

We propose a method for jointly predicting a segmentation map and tracking surgical instruments in two types of interventions: robotically assisted endoscopic surgery and retinal microsurgery. An overview of the method is shown in Figure 6.1.

Due to the challenges associated with minimally invasive surgeries, such as hand-eye coordination or the reduced field-of-view, computer-aided systems have been developed to support the surgeon. In such systems, estimating the exact position and orientation of surgical tools is a crucial component. One non-trivial specification of intra-operative systems based on computer vision is *robustness* in handling challenging conditions such as strong illumination changes, occlusions, smoke or motion blur [44]. Another important consideration is that such systems must operate in *real time*. Thus, accurate localization and tracking of surgical instruments has been an active topic of research in the medical/vision community [7, 43, 44, 89, 219, 327, 340, 344, 345, 364, 395, 397, 440, 486, 487].

In previous work, segmentation and localization have been often addressed as two independent tasks, with one following the other in a serial manner. In fact, segmentation has been used both as a prerequisite for instrument localization in [6] and as a post-processing step to improve an initial localization estimate in [341]. This suggests, as is also intuitive, that these two tasks are highly correlated and we hypothesize that jointly learning both tasks can bring both a computational speed-up and a performance boost.

6.2.1 Methodology

Next we present our approach for joint segmentation and localization of surgical tools as a multi-task adaptation of the *fully convolutional architecture* presented in Chapter 4 and discuss why addressing localization as high-dimensional image-to-image mapping is beneficial in this setting.

We refer to segmentation as the pixel-wise labeling of the entire scene, either binary (foreground/background) or semantic (into different parts of the tool, such as shaft, manipulator, etc. or different tools). We refer to localization as the estimation of the 2D coordinates (on

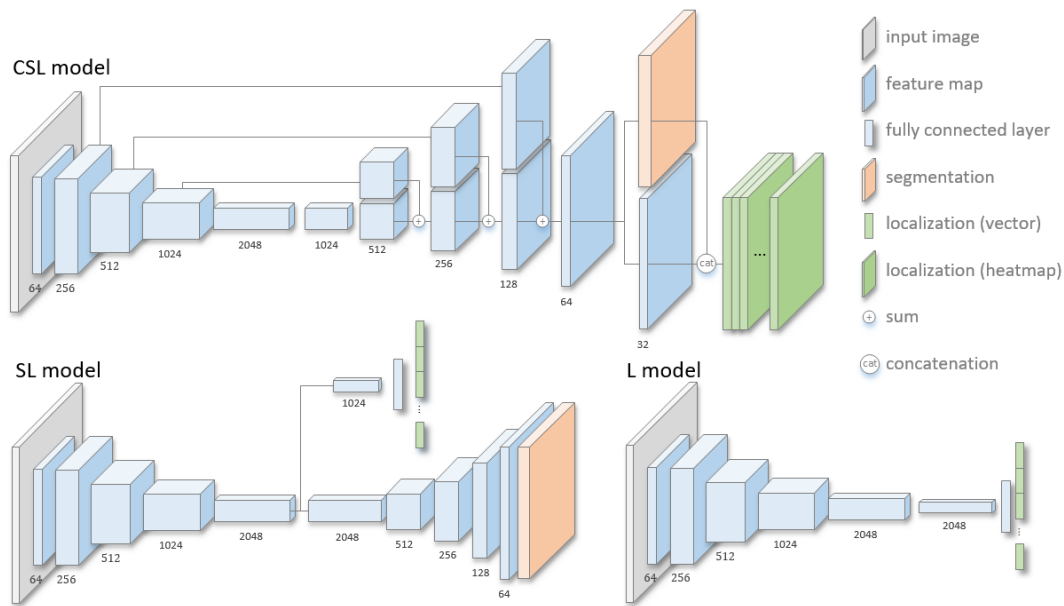


Figure 6.2 Concurrent segmentation and localization (CSL) architecture. We also show two baselines: the single-task localization-only model (L) and the multi-task localization-and-segmentation model (SL) where only encoder weights are shared among the two tasks. Instead, our proposed architecture CSL treats the two tasks with the same output dimensionality and all weights in the network (except for the prediction layers) are shared.

the image plane) of specific landmarks of the articulated tool, as for example seen in Figure 6.1. Tracking is essentially achieved by continuously detecting the tool landmarks in each frame.

Let (I, S, k) be a training sample as, where $I \in \mathbb{R}^{W \times H \times 3}$ is the input image, and correspondingly, $k \in \mathbb{R}^{(N \times 2)}$ represents the annotated 2D coordinates of N landmarks (keypoints) and $S \in \mathbb{R}^{W \times H \times C}$ is the ground truth segmentation into C labels. W and H correspond to the image width and height respectively.

In the following, we discuss our approach and two baseline architectures, which are all illustrated in Figure 6.2. In all variants, the image encoder remains the same as in Chapter 4, that is ResNet-50 up to `res5c`, without the final pooling and fully connected layers. This backbone allows for real-time computation during inference, which is a requirement for intra-operative procedures. We investigate the following three problem formulations and discuss the decoder layers adapted for each scenario.

Localization (L)

We begin with a simple baseline of directly regressing the 2D coordinates of the tool landmarks (ignoring the segmentation task). Thus, the setting here is single-task. We use the image encoder described above and append one additional residual block with stride to further reduce the spatial dimensions of the feature maps. The residual block is then followed

by an average pooling layer and a fully connected layer that produces an output $\tilde{\mathbf{k}} \in \mathbb{R}^{2N}$ (reshaped to $\mathbb{R}^{N \times 2}$). We train the model with \mathcal{L}_2 loss to directly regress image coordinates.



NOTE: *Localization precision*

Intuitively (and experimentally verified), the accuracy of this model improves when reducing the resolution of the feature maps through the additional residual block instead of using a larger window when average pooling. Large average pooling windows introduce spatial invariance that negatively affects the network’s ability to localize precisely.

Segmentation and Localization (SL)

We then extend the setting to multi-task and predict 2D landmark coordinates and a segmentation map simultaneously with a single network. To do so, the weights of the encoder are shared but the architecture splits into two branches after `res5c`. Each branch is task-specific and features a different output dimensionality. The localization branch follows the design described above (L), while the segmentation branch consists of four residual up-sampling layers, exactly as described in Section 4.3, predicting a pixel-wise probability distributions over classes $\tilde{S} \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times C}$. As in Section 4.4, the segmentation objective is softmax cross-entropy and the total loss becomes:

$$\mathcal{L}_{SL} = \mathcal{L}_{CE}(S, \tilde{S}) + \lambda \mathcal{L}_2(\mathbf{k}, \tilde{\mathbf{k}}), \quad (6.1)$$

where λ is a factor to balance the contribution of the two terms.

Concurrent Segmentation and Localization (CSL)

Often, landmark locations cannot be annotated with painstaking precision as their exact position might be ambiguous in an image of high resolution; most of the time, there exists a small image region in which the annotation can be still considered valid. Despite this fact, the previous two baselines are optimized against the annotated 2D coordinates, therefore attempting to minimize the distance of a predicted location to an ambiguous annotation.

Recent work in the field of human pose estimation [35, 49, 409, 438] shows that it is preferable to estimate joint positions by regressing a set of heatmaps, one for each joint, instead of predicting the exact pixel location. Each heatmap is essentially a confidence map, representing the likelihood of the joint occurring in a spatial location. This approach seems more natural than direct regression and allows to model the spatial ambiguity of the annotation.

Inspired by this, we model the problem of surgical tool localization in a similar way. This further allows to lift both tasks to the same dimensionality and share weights not only across the encoder but also across the decoding layers. This, in turn, achieves greater computational efficiency comparing to the SL-model and, as the tasks of segmentation and localization are highly correlated, we expect performance to also benefit from the extended sharing of parameters.



NOTE: *Multi-task learning and parameter sharing*

Baxter [32] shows that shared parameters (or a common learned representation) can lead to improved generalization, when the learning tasks are similar. This might not be the case among tasks that are competitive. When sharing is justified—and we believe this is the case here—the shared part of the model in a multi-task setting is constrained towards values that generalize better [128].

We can now use FCRN, which is fully shared across both tasks. We also enhance the architecture by introducing *long-range residual connections* between the encoder and decoder layers, as shown in Figure 6.2. These connections enable lower-level (higher-resolution) information passing directly from the initial encoder layers to the upper layers (of corresponding spatial dimensions) through summation of the features. It is shown that information flow through skip connections benefits tasks such as segmentation [347]. A convolution is introduced in the skip connection to adjust that the number of feature maps to that of each decoding stage (the numbers are shown in the same figure). The two tasks are split only at the prediction layers of the network which consist of 3×3 convolutional layers. For the segmentation branch we append a convolutional layer with C filters to the last up-projection layer and predict the segmentation class probabilities (softmax-ed) by optimizing \mathcal{L}_{CE} . For the localization branch, we first append a convolutional layer of 32 filters to which we concatenate the output of the segmentation branch, thus using segmentation to guide the heatmap regression. This is followed by ReLU activation and another convolution of N filters, which yields the dense localization output $\mathcal{H} \in \mathbb{R}^{\frac{W}{2} \times \frac{H}{2} \times N}$ supervised by:

$$\mathcal{L}_H(k, \mathcal{H}; \sigma) = \frac{1}{NHW} \frac{1}{\sqrt{2\pi\sigma^2}} \sum_{n=1}^N \sum_{i=1}^H \sum_{j=1}^W \left(\exp\left(-\frac{\|k_n - (i, j)\|_2^2}{2\sigma^2}\right) - \mathcal{H}_{i,j,n} \right)^2. \quad (6.2)$$

In Equation (6.2) we have reformulated the localization target by placing a 2D Gaussian distribution with its peak at the annotated landmark location k_n and constant variance σ^2 which controls the spread of the Gaussian around the peak. This is done for each landmark independently. The overall objective is then given by:

$$\mathcal{L}_{CSL} = \mathcal{L}_{CE}(S, \tilde{S}) + \lambda \mathcal{L}_H(k, \mathcal{H}; \sigma) \quad (6.3)$$

During inference, the tracked landmark location is the point of maximum confidence in the corresponding predicted heatmap. When the model is uncertain (*e.g.* because of motion blur) or if detection failure occurs, the predicted heatmaps show very high variance.

6.2.2 Results and Evaluation

We evaluate our method in two settings: in retinal microsurgery and endoscopic surgery. These differ in the anatomy that is being operated, the tools used and the challenges involved in the operation itself.

Datasets

The Retinal Microsurgery (RM) dataset [344] consists of 18 in-vivo sequences, each with 200 frames of resolution 1920×1080 pixels, recorded at 25 fps. There are four different instruments used in total over all sequences, which splits the dataset into four instrument-dependent subsets. Provided are annotations of $N = 3$ tracked landmarks on the tooltips and binary segmentations tool/background ($C = 2$).

For the endoscopic surgery we use the data from the EndoVis 2015 challenge. This is a challenge designed to evaluate segmentation and tracking methods for robotic instruments in laparoscopic surgery. The training samples correspond to four ex-vivo 45s-long sequences, while as testing samples the remaining 15s of the same sequences are used. There exist two additional 60s sequences in the test set. The resolution of the sequences is 720×576 pixels. There exist one or two (simultaneous) surgical instruments over all sequences, with only $N = 1$ annotated landmark per instrument. The semantic classes are $C = 3$: manipulator, shaft and background.

Implementation Details

All frames are first resized to 640×480 pixels, although as input to the network we choose a resolution of 480×480 . The following augmentations are randomly sampled and applied online, during training:

- Rotations of $[-5, 5]$ degrees.
- Scaling with a factor of $[1, 1.2]$.
- Random crops of 480×480 pixels.
- Gamma correction with $\gamma \in [0.9, 1.1]$.
- Multiplicative color transformations with a factor $c \in [0.8, 1.2]^3$.
- Specular reflections (in EndoVis).

For the 2D Gaussians, we set $\sigma = 5$ for the RM dataset and $\sigma = 7$ for EndoVis in which the surgical instruments are larger. We train all discussed models after initializing ResNet-50 weights from the ImageNet pre-trained models. We train with stochastic gradient descent with a constant learning rate of 10^{-7} and momentum 0.9. The weight factor λ between localization and segmentation objectives in SL and CSL is set to 1.

Evaluation Metrics

We evaluate the segmentation results using the following metrics (TP: true positives, FP: false positives, TN: true negatives, FN: false negatives):

- DICE coefficient (binary case): $\frac{2TP}{2TP + FP + FN}$
- Recall (Sensitivity): $\frac{TP}{TP + FN}$
- Specificity: $\frac{TN}{TN + FP}$

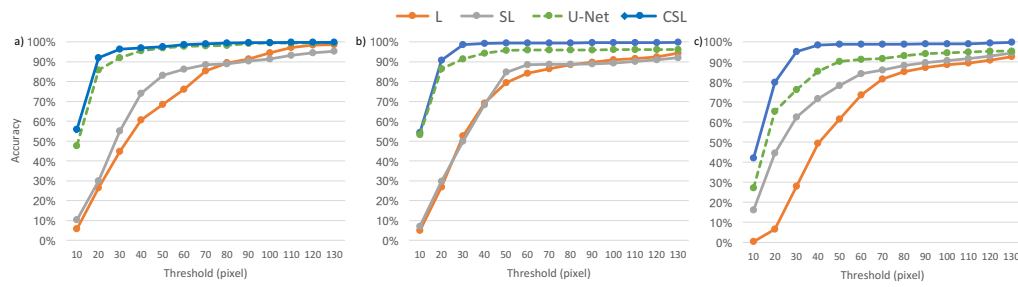


Figure 6.3 Ablation experiments on the Retinal Microsurgery dataset. We plot and compare the localization accuracy of the baseline models (L, SL), U-net [347] and the proposed method (CSL) for (a) the left tip, (b) the right tip and (c) the center joint of the instrument.

The DICE score is similar to the Jaccard index. Recall measures the amount of positives that are correctly classified, while specificity measures the amount of negatives that are correctly classified.

Localization is evaluated using metrics from previous work [345, 395]. If for each landmark, $k_n \in \mathbb{R}^2$ are the coordinates of the ground truth and $\tilde{k}_n \in \mathbb{R}^2$ the corresponding prediction (with $n = 1, \dots, N$ used as the landmark index), then we define the following metrics:

- Threshold score: $\|k_n - \tilde{k}_n\|_2 < T$, where $T \in \mathbb{R}$ is a fixed threshold
- Keypoint Threshold Bounding Box (KBB): $\|k_n - \tilde{k}_n\|_2 < \alpha \max(h, w)$, where $\alpha \in \mathbb{R}$ and h, w are the height and width of the ground truth bounding box around the instrument.

Ablation Studies

We evaluate and compare all models (L, SL, CSL) on the RM dataset. We train on 9 sequences and test on the remaining ones. We plot the results in Figure 6.3 for the localization accuracy of the left, right and center tip of the instrument over different thresholds. The localization-only baseline (L) shows the weakest performance. With the SL architecture, however, we already see an improvement in localization performance, especially for strict thresholds, which is solely thanks to the multi-task setting. We then evaluate the potential of localization via heatmaps with the CSL architecture; the performance of this model surpasses the baselines by a significant margin, reaching 90% accuracy for the right and left tool tips and 79% for the center joint (for threshold equal to 20 pixels). The performance gain here can be attributed to both the improved multi-task supervision setting and the improved formulation of the localization problem.

We also compare our model to the U-net [347], which is an architecture broadly used in medical applications, especially for segmentation. We train U-Net with the same two objectives as CSL. CSL consistently achieves higher localization accuracy and, as shown in Table 6.1, higher segmentation accuracy as well.

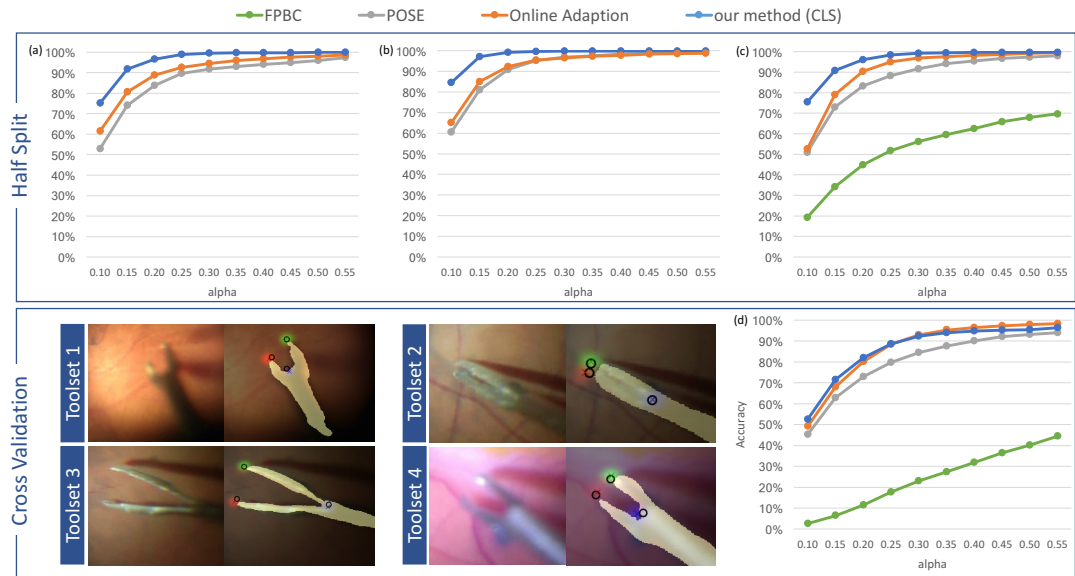


Figure 6.4 Comparisons on RM dataset. We compare our method to FPBC [396], POSE [344] and Online Adaption [345]. Half Split: (a) to (c) correspond to the KBB scores of left tip, right tip and center joint. Cross Validation: (d) shows the average KBB score for the center point over all folds.

Architecture	DICE
U-net [347]	0.725
SL	0.737
CSL (w/o skip connections)	0.744
CSL	0.754

Table 6.1 Segmentation accuracy on the Retinal Microsurgery dataset. We compare the DICE scores of U-net [347] and baselines SL and CSL without long-range skip connections to the proposed model (CSL).

State-of-the-art Comparisons

Retinal Microsurgery In order to fairly compare to previous work, we train on RM using the first half of each of the 18 sequences and evaluate on the remaining half of the frames. Quantitative results are shown in Figure 6.4, where we refer to this split as *Half Split*. In addition, to evaluate the generalization ability of our method to unseen *geometry*, *i.e.* novel instruments, we follow a leave-one-out cross-validation scheme on the subsets defined by the 4 different instrument types. We refer to this experiment as *Cross Validation* and the results are also shown in Figure 6.4.

EndoVis 2015 Following the guidelines for this challenge, we carry out leave-one-out experiments and evaluate on the last 15s of the left-out sequence (for sequences 1–4). For test sequences 5 and 6 we train on all available data. There is an additional challenge in this dataset as there are sometimes two instruments present in the testing sequences (one from the left and one from the right), while only a single tool (right) is present throughout the training data. We deal with this challenge by augmenting the training data with ran-

Sequence	Segmentation								Localization	
	Binary				Shaft		Grasper		Tool 1	Tool 2
	B.Acc	Rec	Spec	DICE	Rec	Spec	Rec	Spec	RMSE	RMSE
1	91.9	85.0	98.7	88.5	79.2	99.1	76.2	98.7	39.0	30.8
2	94.8	90.0	99.7	93.0	90.9	99.8	82.0	99.8	9.7	N/A
3	94.7	90.1	99.3	91.6	89.1	99.5	86.8	99.7	10.9	N/A
4	91.1	83.1	99.0	85.8	82.9	99.2	65.4	99.6	13.0	N/A
5	91.5	84.2	98.8	87.3	82.8	99.1	75.9	99.2	38.4	60.0
6	91.7	84.9	99.0	88.9	78.0	99.3	78.1	98.4	36.4	63.9
CSL (mean)	92.6	86.2	99.0	88.9	83.8	99.3	77.4	99.2	24.8	51.6
FCN [115]	83.7	72.2	95.2	-	-	-	-	-	-	-
FCN+OF [115]	88.3	87.8	88.7	-	-	-	-	-	-	-
Pakhomov et al. [313]	92.3	85.7	98.8	-	-	-	-	-	-	-

Table 6.2 Quantitative comparisons on EndoVis 2015. We evaluate segmentation using recall (Rec), specificity (Spec), balanced accuracy (B.Acc) and DICE scores. Binary evaluation is done after merging the two semantic classes (shaft and grasper). We also report the localization error for the two tools as the Euclidean distance between the predicted and the ground truth center joint of each tool.

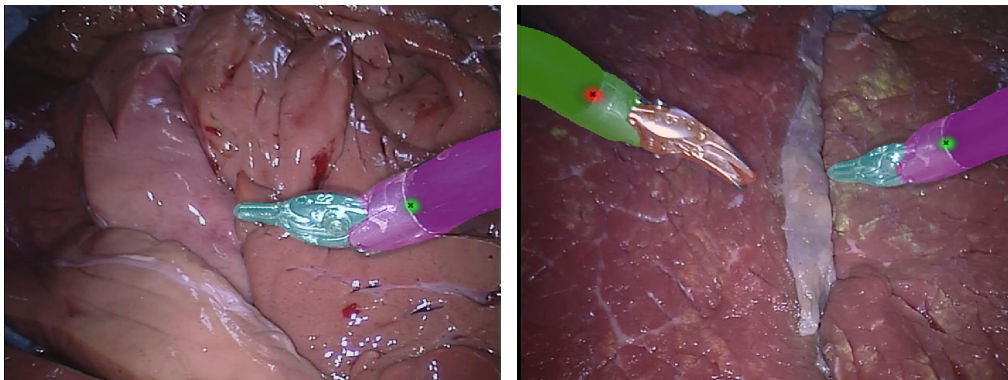


Figure 6.5 Qualitative results on EndoVis 2015. The semantic segmentation (manipulator/shaft) and joint localization are overlaid on the original image. Left: We show a representative frame from a test sequence containing one tool. Right: Test sequence contains two tools, although only the right tool exists in the training data. Due to proper augmentation during training, the model is able to generalize well.

dom horizontal flipping, which makes at least one instrument seen from both sides. When flipping, we change the corresponding semantic labels, which allows for multi-tool segmentation into $C = 5$ classes: left shaft, right shaft, left grasper, right grasper and background, *i.e.* we are able to distinguish left and right instruments. Additionally, we can differentiate between the left/right joint locations with $N = 2$ heatmaps. However, as the evaluation protocol of the challenge supports only 3 classes, we then merge the left/right grasper and and left/right shaft for reporting the quantitative results.

Another challenge of this dataset is strong specular reflections which cause the segmentation to fail for some frames. We have addressed this issue by artificially modeling specular reflections along the shaft during training, thus encouraging the model to be more robust to this condition.

In Table 6.2 we report the segmentation and localization results and compare to previous and concurrent work that also reports their performance on this challenge. In Figure 6.5 we show two qualitative samples from our method, including the challenging scenario of segmenting a previously unseen instrument, which is tackled successfully by our model.

**NOTE:** *EndoVis 2017*

We have also used the proposed architecture and training scheme to participate in the EndoVis 2017 challenge for semantic segmentation. Our method achieved the first place in multi-class segmentation and third overall. The results of the challenge are published in [5].

6.3 Localization of Grasping Points

Robotic grasp detection from visual data is another application of image understanding with an object-centric character, that we explore in this chapter. Related to localization of surgical instrument joints, one of the goals of robotic manipulation is to localize possible grasping points *around* objects in a scene (instead of *on* the manipulator itself). As the nature of the problem is quite similar, it is possible to leverage the same problem formulation as before.

Because of its vast applicability, the task of visual recognition in robotic grasping has attracted a lot of research interest over the years. Some — especially earlier approaches rely on primitive shapes and known object geometry to estimate the grasp position [290, 291, 474]. We refer the reader to the review of Bicchi and Kumar [38] for a more detailed overview of past work on this topic. In the past, knowledge of the 3D model of a given object was essential, due to the lack of RGB-D sensors or the incomplete depth measurements of stereo cameras. However, nowadays, object geometry can be easily estimated, even for previously unseen shapes, given enough training data, as we show in Chapter 4. Most recent methods address robotic grasping via learning techniques (deep or not) [20, 67, 136, 184, 193, 218, 235, 237, 274, 290, 337, 367, 417, 422, 436]. When estimating grasp configurations from visual data, the localization is usually done in image coordinates, which can then be converted into real world coordinates given additional information. Then, given an estimated grasp configuration the robot can proceed to planning and control.

Very common among related methods that estimate grasping points directly from visual data is the use of a 5-dimensional representation, *i.e.* an oriented rectangle that represents the grasp configuration. Because this representation is similar to the one used in object detection [123, 124, 338, 342], some methods address this task by predicting candidate bounding boxes around graspable locations. We take a different approach by reformulating the problem again into heatmap regression, which allows us to model spatial ambiguities in the configuration. We then make an interesting observation; as most objects can be grasped in various ways, optimizing towards a specific grasp per object during training can be harmful. We thus take a step further and model the ambiguous nature of this localization problem by predicting multiple grasp hypotheses per object.

6.3.1 Methodology

In [184, 235] robotic grasp detection is formulated as estimating the size, position and orientation of a 2D rectangle, which results in a 5-dimensional representation (x, y, θ, h, w) , where (x, y) is the center of the rectangle of height h and width w and θ is its orientation relative to the horizontal axis. The height and width depend on the gripper. The representation above is suitable for modelling a parallel plate gripper and is very frequently used in related work [20, 136, 218, 235, 337, 436].

Heatmap Representation

Similar to the example of surgical tools, we can model and localize several joints on the end-effector following our heatmap formulation, which can be extended to N -finger grippers. In the general case, for an N -finger gripper, the heatmap can be constructed as a mixture model of N bivariate Gaussian distributions fitted around each finger location. It is important to note that here, although the end-effector is not part of the scene, it is implied that the grasp (*i.e.* the localization outcome) is immediately related to the configuration of the gripper.



NOTE: *Heatmap as a Gaussian mixture*

In Section 6.2 we have used one heatmap per joint, whereas here we use a combined heatmap for both grasp points via a Gaussian mixture model. The reasoning behind this is twofold. When we estimate the pose of tool joints, where the tools are actually visible in the image, their identity can be unambiguously inferred. Here, the gripper — the joints/fingers of which we are implicitly localizing — is not visible so its orientation cannot be inferred visually. The second reason is that many grippers are actually symmetric or can rotate 360° , thus identifying the permutation of the two grasp points is irrelevant.

We reformulate the oriented rectangle representation as:

$$G(\mathbf{p}) = \sum_{n=1}^N \frac{\exp\left(-\frac{1}{2}(\mathbf{p} - \boldsymbol{\mu}^{(n)})^T \mathbf{R}(\theta) \boldsymbol{\Sigma}^{-1} \mathbf{R}(\theta)^T (\mathbf{p} - \boldsymbol{\mu}^{(n)})\right)}{\sqrt{2\pi N} \sigma_x^{(n)} \sigma_y^{(n)}}. \quad (6.4)$$

In the above equation, \mathbf{p} denotes pixel indices. The Gaussian distributions are parametrized by the specifications of the gripper fingers. We define the means $\boldsymbol{\mu}^{(n)} = (\mu_x^{(n)}, \mu_y^{(n)})^T$ with $n \in \{1, \dots, N\}$ as the 2D centers of the gripper plates (in image coordinates). In the case of the parallel gripper, the distance of the means $\|\boldsymbol{\mu}^{(1)} - \boldsymbol{\mu}^{(2)}\|_2$ is equivalent to the width (aperture) w of the gripper. In contrast to the heatmaps of Section 6.2.1 which have a constant variance, here, we create ellipsoidal Gaussian distributions with $\boldsymbol{\Sigma} = \text{diag}(\sigma_x^{(n)}, \sigma_y^{(n)})^2$ such that the $\sigma_x^{(n)}$ corresponds to the height (length) h of the gripper plates (while $\sigma_y^{(n)}$ is constant). Further, the mixture model is oriented by rotating the heatmap with rotation matrix $\mathbf{R}(\theta)$. A graphical representation of this adaptation is shown in Figure 6.6. In Figure 6.7 we show the annotated grasp configurations for an image from the Cornell grasp detection dataset [235] along with the heatmaps we generate using the ground truth rectangles.

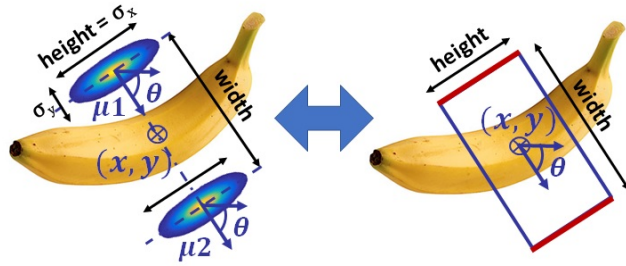


Figure 6.6 Reformulating oriented rectangles as heatmaps. The heatmaps are created as Gaussian mixtures with the plate centers as means and the variance σ_x proportional to the gripper height (σ_y is a chosen constant).

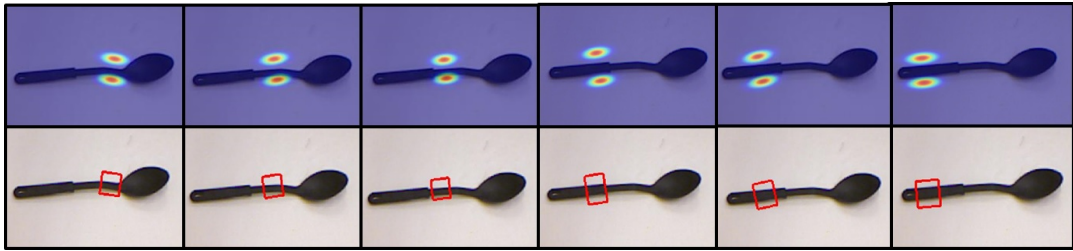


Figure 6.7 Multiple grasp configurations for an object. We show all annotated oriented rectangles for an image of the Cornell grasp detection dataset [235] and the corresponding heatmaps (our representation).

Our representation holds the same information as the oriented rectangles used in prior work and, in addition, it expresses the spatial uncertainty around a grasp. Thus, for a model trained to predict grasp heatmaps, the likelihood of a predicted grasp is reflected in the magnitude and variance of the predicted heatmap. For this reason, we find this a more interpretable representation for this task and more easily generalizable to multi-finger grippers comparing to the oriented rectangles.

Multiple Grasp Hypotheses

We train a FCRN architecture (Section 4.3.2) to learn the mapping from RGB images to grasp heatmaps. Since we now represent the joints as mixture models, the output heatmap consists of just a single channel, *i.e.* we predict $\tilde{G} \in \mathbb{R}^{\frac{W}{2} \times \frac{H}{2}}$, where W, H are the width and height of the input image.

However, we are now facing the following challenge regarding the supervised training of this model. For most objects there are several grasping possibilities and it is unlikely that there is only a single configuration that leads to a successful grasp. This is also reflected in related datasets, which include a variable number of grasp annotations per object (an example shown in Figure 6.7). Ideally, we would like to learn from the distribution of grasp locations rather from a single ground truth example. If we optimize the model towards a pre-selected ground truth grasp per object, then we might penalize predictions that are presumably valid grasping possibilities. In this case, the samples that the model learns from

do not cover the entire distribution of grasps. This can make the model overfit to the specific choices in the training set, which reduces its generalization capabilities.



NOTE: *Related work*

Most of the related work does not address the ambiguity of robotic grasping. In [337], a random ground truth is chosen at each training step and treated as a single valid grasp. However, more recently and heavily inspired by Faster R-CNN [342] for object detection, Chu et al. [67] predict and classify several grasp proposals per object.

As we show later supervising our model with exactly one grasp — the most stable annotated grasp per object — yields results that are far from optimal. Supervising with a randomly selected grasp per training step is even worse, because the model learns the conditional average of the grasp distribution, which is likely to fall in-between modes, often resulting in an unsuccessful grasping outcome.

The way we propose to address this ambiguity in robotic grasping has its theoretical foundation in our previously published work, which we refer to as Multiple-Hypothesis Prediction (MHP) [357]. A common characteristic of previous methods in this field is addressing ambiguity in the predictions of a model by allowing the model to produce multiple outcomes [107, 144, 145, 178, 233, 247, 357]. This is necessary in tasks that might have several plausible outcomes instead of a single definitive answer; for example, future prediction [107, 357], pose estimation [279, 357], segmentation [145, 247, 357], optical flow [178] or image captioning [145]. This is also the case here, although multiple-hypothesis learning has not been yet investigated in the context of robotic grasping.

Our solution is to adapt the model to predict multiple grasp hypotheses simultaneously:

$$\tilde{G} = \{\tilde{G}^{(m)}\}, m \in \{1, 2, \dots, M\}, \quad (6.5)$$

where M is a hyper-parameter denoting the number of hypotheses. This can be done by replicating the last layer M times, *without* weight sharing.

As in MHP frameworks, the model is trained by minimizing an *oracle meta-loss* $\mathcal{M} : \mathbb{R}^M \rightarrow \mathbb{R}$, which is a function of a task-dependent loss \mathcal{L} . We independently compute the cost of each output against the same ground truth G . Then, the oracle meta-loss can be interpreted as a winner-takes-all loss as it selects the output with the minimum cost, that is

$$\min_{m=1, \dots, M} \{\mathcal{L}(\tilde{G}^{(m)}, G)\}. \quad (6.6)$$

It arises that the loss for a given sample is the loss of the currently *best* hypothesis. When training by back-propagation only the selected hypothesis will be updated at this step. However, this formulation can cause some hypotheses that did not receive updates early enough during training to be weak, which in turn makes it less and less likely that they are selected as the best hypothesis later on, thus causing them to collapse. As we discuss in [357], it

is possible to overcome this issue with a soft approximation of the minimum loss and we define \mathcal{M} as:

$$\mathcal{M}(\tilde{G}, G^*) = \left(1 - \epsilon \frac{M}{M-1}\right) \min_{m=1, \dots, M} \{\mathcal{L}(\tilde{G}^{(m)}, G^*)\} + \frac{\epsilon}{M-1} \sum_m \mathcal{L}(\tilde{G}^{(m)}, G^*), \quad (6.7)$$

where ϵ is a small scalar ($\epsilon = 0.05$). In the above equation, the oracle selects the best hypothesis with weight $1 - \epsilon$, but allows small updates for all other hypotheses as well with a shared factor of $\frac{\epsilon}{M-1}$.

In most ambiguous problems addressed by previous work, there is often only a single annotated ground truth. In Equation (6.6) we also make the assumption that there is a single annotated ground truth G against which we compare all predictions. However, when several annotations are available for the same image (as is the case in [235]), we can randomly sample a target heatmap $G^* \sim \mathcal{G}$ (as in Equation (6.7)) among all available grasps \mathcal{G} for a given image every time it is seen during training. At the end of training, all hypotheses are expected to be trained equally and, given that multiple target grasps have been now encountered during training, the hypotheses should model the real distribution of grasps.

Hypothesis Selection

If such model is to be integrated in a real robot, then being able to select a good grasp among all hypotheses is vital. In the field of multiple-output learning, some methods propose selection algorithms to choose one of the predicted hypotheses as the output of the system [144, 247]; these are usually task-dependent.

Here a logical approach would be to reverse the process of generating grasp heatmaps. By construction, the model is trained to predict a mixture model of multivariate Gaussian distributions. In order to rank the predictions, we propose to fit a two-component Gaussian mixture to each hypothesis using finite mixture model estimation [288].

A mixture model of multivariate distributions can be defined as

$$p(\mathbf{x}) = \sum_{k=1}^K \phi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \quad \sum_{k=1}^K \phi_k = 1, \quad (6.8)$$

where ϕ_k , $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the mixing coefficient, mean and covariance matrix of component k respectively. By determining the maximum likelihood of this model, we can find the optimal values for these components. As estimating the maximum likelihood analytically is intractable, we optimize the fit via the Expectation-Maximization (EM) algorithm [79].

NOTE: EM algorithm

EM is an iterative method to find a numerical solution to the maximum likelihood problem. There are two main steps: (E) computes an expectation of component assignments for each given data point given the current parameters and (M) computes a maximum likelihood estimation and subsequently updates the model parameters. The algorithm iterates over these two steps until the error is below some threshold.

When the fitting has converged, each hypothesis will be assigned a likelihood which can be interpreted by how well it resembles a Gaussian mixture. We use the estimated likelihoods to rank the hypotheses and choose the top-ranked one as the system's output.

6.3.2 Results and Evaluation

We evaluate our approach on the Cornell Grasp dataset [235], that is a commonly used benchmark for this task. Our point of focus is the benefit of a multiple-prediction versus the single-prediction setting, but also the hypothesis selection criterion.

Dataset

The Cornell dataset [235] consists of 885 RGB-D images from 240 graspable objects with a resolution of 640×480 pixels. Some samples from this dataset are shown in Figure 6.8. There multiple grasping possibilities annotated as oriented rectangles for each object (Figure 6.7), mostly assuming 2D grippers with parallel plates. There are 2 to 25 grasp rectangles per object in different scales, orientations and locations, but the labels are not exhaustive.

Previous work [20, 136, 218, 235, 337, 436] uses two different splits as the evaluation protocol, with cross-validation for each split (five folds each). The *image-wise* data split contains all possible objects in the training set, while the validation set consists of unseen views of the *same* objects. This is to evaluate the intra-object generalization of the methods. The *object-wise* split contains all views of the same object in the training set and *novel* objects in the validation set. However, unseen objects are very similar in shape to the ones used for training. For example, there are several objects of different colors but of similar shape. Thus, the overall performance of methods does not vary much compared to the image-wise split.

We believe that the object-wise split might not be representative of a method's generalization capabilities to novel shapes, although the shape of an object is the most important factor in grasping. In order to study our model's performance on truly novel shapes, we propose an additional *shape-wise* split to ensure a larger variation in objects between train and test set. We suggest a two-fold cross-validation experiment on the more challenging shape-wise split with 20% of the objects used for validation in each fold. Here the validation set is constructed from objects that are most dissimilar to the ones used for training.

Implementation Details

We train the model after offline augmentations on the dataset. The transformations we apply are the following:

- Rotation with an angle of $[-60, 60]$ degrees.
- Scaling with a factor of $[0.9, 1.1]$.
- Center crops with an offset of $[-20, 20]$ pixels.

Each image is augmented six times by randomly sampling from the above transformations, which results in a total of 5310 training images after augmentation.



Figure 6.8 The Cornell Grasp dataset [235]. We show some sample (center-cropped) images from the dataset.

As the original images contain a lot of background around the objects, we crop them and the corresponding ground truth heatmaps to 350×350 pixels and then bilinearly down-sample the image to 256×256 and the ground truth to 128×128 (since the output of the network is only half of the input resolution).

When training the single-prediction baseline we choose the most stable grasp as supervision, which is defined as the largest annotated grasp rectangle. When training the multiple-prediction model, we randomly sample from the available ground truth heatmaps per image. We optimize with stochastic gradient descent and a learning rate of 0.0005, momentum of 0.9 and weight decay of 0.0005. We train for 50 epochs and a batch size of 5 and 20 for training multiple and single prediction models respectively. For the MHP models, we also introduce hypothesis dropout as regularization with a rate of 0.05, *i.e.* we randomly drop a hypothesis with 5% chance. This is another step to ensure that a single hypothesis will not be over-trained while others are never selected and collapse. For evaluation, we use the top-ranked prediction after optimizing with EM for 1000 iterations.

Error Metric

We evaluate our results using the metric suggested by [184], which is also used in previous work. According to this metric, we define grasp success when

- The intersection over union (IoU) between the ground truth bounding box and the predicted bounding box is greater than 25%, and

Method	Input	Grasp Estimation Accuracy (%)			
		Image-wise	Object-wise	Shape-wise	
Lenz et al. [235]	RGB-D	73.9	75.6	-	
Wang et al.[436]	RGB-D	85.3	-	-	
Redmon et al.[337]	RGB-D	88.0	87.1	-	
Asif et al.[20]	RGB-D	88.2	87.5	-	
Kumra et al.[218]	RGB-D	89.2	89.0	-	
Guo et al.[136]	RGB-D, tactile	93.2	89.1	-	
Kumra et al.[218]	RGB	88.8	87.7	-	
<i>single</i>	M = 1	RGB	83.3	81.0	73.7
<i>multiple</i>	M = 5	RGB	91.1	90.6	85.3
<i>multiple (diversity)</i>	M = 5	RGB	89.1	89.2	82.5
<i>multiple</i>	M = 10	RGB	91.5	90.1	86.2

Table 6.3 Comparison with the state of the art on grasp detection accuracy. Single predicts a single heatmap, while multiple refers to our MHP models. The top-ranked hypothesis is evaluated for the MHP models.

- The difference in orientation between the ground truth rectangle and the predicted rectangle is less than 30° .

When these conditions are met towards *any* of the available ground truth rectangles, then the predicted grasp is considered successful. We compute the percentage of successful grasps, which we refer to as *Grasp Estimation Accuracy*.

It is clear that in order to compute this metric, we first need to convert the predicted heatmap back to an oriented rectangle representation. The heatmaps are thresholded with $t > 0.2$. The Euclidean distance between the means μ_1 and μ_2 of each Gaussian correspond to the rectangle’s width w . The major axis of the ellipse corresponds to the height h of the rectangle. Finally, we compute the orientation θ as the angle of the major axis as $\arctan\left(\frac{d_1}{d_2}\right)$, where d_1 and d_2 are the vertical and horizontal distances between the centers of the two Gaussians respectively.

Sometimes, when the variance of a predicted heatmap is high, we discard the hypothesis as a rectangle cannot be reliably extracted. Thus, variance is often a prognostic of grasp quality.

Quantitative Evaluation

We train models for $M = 5$ and $M = 10$ and compare them to the single-prediction baseline ($M = 1$). The results are shown in Table 6.3. When modeling ambiguity with multiple hypotheses we observe a significant improvement over the baseline. Increasing the number of hypotheses from 5 to 10 shows a smaller improvement. Although we only use RGB data as input to the model, our approach surpasses the performance of previous methods

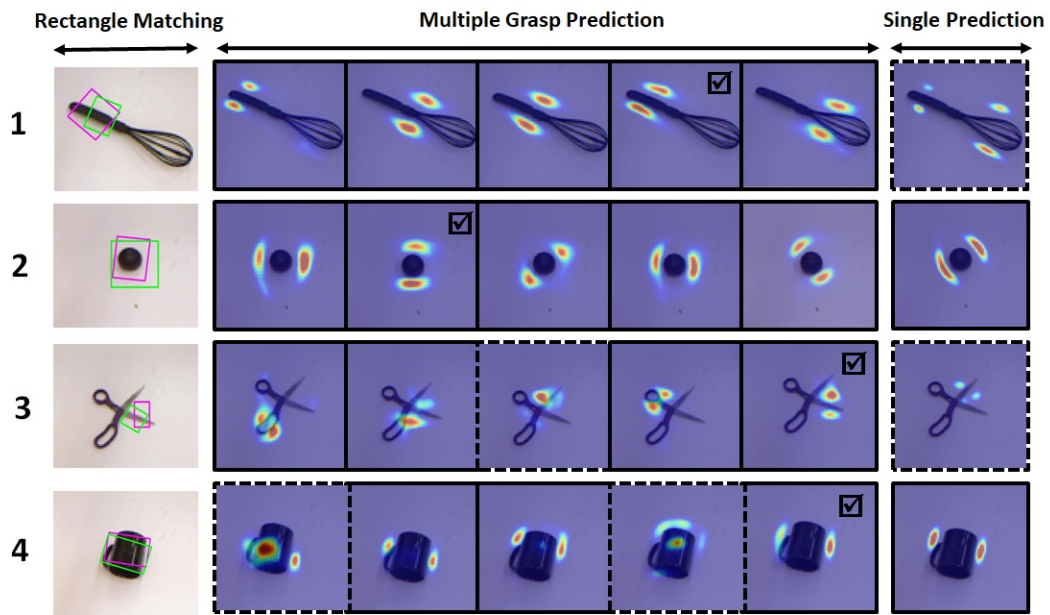


Figure 6.9 Predicted heatmaps from our model with $M = 5$. We also compare to the single-prediction model (right). A solid frame around heatmaps indicates a successful grasp, while a dashed line indicates a missed detection. (✓) indicates the top-ranked hypothesis. The rectangles resulting from top-ranked heatmaps are overlaid on the RGB image (magenta) against the closest ground truth (green).

that make use of RGB-D data (depth information usually adds a few points of performance). Concurrently to our approach, some methods have extended Faster R-CNN [342] to grasp detection resulting in superior performance [67, 489]. As we discuss later, the small performance gap could be attributed to our selection mechanism, which does not always choose the best hypothesis (Table 6.4).

We further report performance under the more challenging shape-wise split to better evaluate generalization to novel objects. In this case, the baseline accuracy drops by nearly 10 points comparing to the image-wise split. On the other hand, the multiple hypothesis models show a smaller decline with respect to the image- and object-wise accuracy. With an increasing number of hypotheses, the performance gap over the single-prediction model is highest for the shape-wise split, with over 12% increase in accuracy.

NOTE: *Over-fitting to the training data*

By proposing a new split on this data, we have attempted to show that models with nearly perfect performance (under certain metrics) might not necessarily adapt well to more challenging real-world scenarios, for example when encountering complex object shapes that are not similar to those in the training distribution. This suggests that to further advance the field, more challenging benchmarks and stricter evaluation protocols might be necessary.

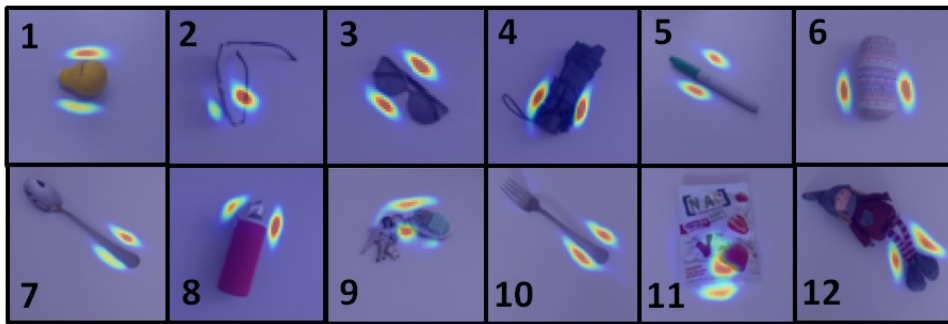


Figure 6.10 Generalization to common household objects. We show the top-ranked heatmap predicted for several self-recorded images. Objects 1-5 are similar in shape to the objects in the Cornell dataset, but objects 6-12 show novel shapes and textures.

Qualitative Results

In Figure 6.9 we show predictions from our $M = 5$ model. We also compare to the single-prediction model. We observe that for objects that can be clearly grasped in variety of ways, our multiple prediction model can disentangle the various outcomes, resulting in diverse predictions that successfully approximate the ground truth distribution. Perhaps the most challenging among these objects are the scissors, for which some of the hypotheses do not result in likely Gaussian mixtures. However, our ranking mechanism can alleviate this issue, by selecting one of the more probable hypotheses.

We also highlight the ability of our model to generalize to real-world scenes (and new objects) by evaluating it qualitatively on self-captured images of common household objects (Figure 6.10).

Diversity of predictions

It is important to understand how diverse the predicted grasp hypotheses are. Since we aim to model the distribution of ground truth grasp configurations, ideally the hypotheses should be representative of that distribution. In objects with many viable grasps are viable, we expect to the model to predict many distinct outcomes. In addition to the heatmaps shown in Figure 6.9, we also present the diversity of the predictions in the form of grasp rectangles in Figure 6.11.

Moreover, we have attempted to encourage more diversity in the predictions explicitly, using a repelling regularizer [346]. The regularizing term is added to the oracle loss with a weight factor of 0.1. We report this experiment in Table 6.3, noted as *multiple (diversity)* ($M = 5$). However, the accuracy of this model is slightly worse than the MHP model without the regularizer. To measure the similarity of the hypotheses in both cases, we compute the cosine similarity among all hypotheses for each test image; the lower the similarity, the better. The average similarity for the object-wise split without the regularizer is 0.435, which is only slightly higher than the respective model with regularizer with a similarity score of

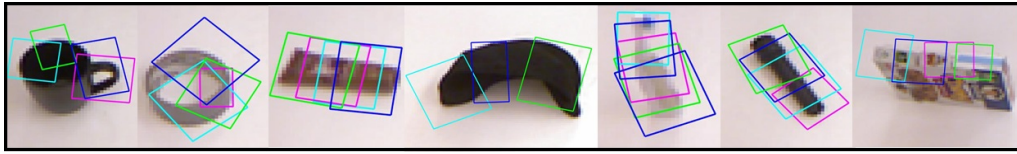


Figure 6.11 **Grasp rectangle prediction.** We show the diversity of the predicted grasp configurations, after they have been converted from heatmaps to oriented rectangles.

Method	Image-wise	Object-wise	Shape-wise
lower limit ($M = 5$)	80.0	77.4	75.0
lower limit ($M = 10$)	76.5	73.3	72.1
upper limit ($M = 5$)	98.0	98.5	96.3
upper limit ($M = 10$)	99.2	98.4	99.1

Table 6.4 **Average grasp estimation accuracy (%) across hypotheses.** For the lower limit we evaluate all hypotheses; for the upper limit any correct hypothesis (oracle) leads to grasp success.

0.427. This suggests that the hypotheses are already sufficiently diverse and do not really benefit from an explicit optimization of diversity.

Evaluation over Multiple Grasps

For a more in-depth evaluation of the grasping hypotheses, we compute the lower and upper limits in detection accuracy and report the results in Table 6.4. The word *limit* here is not used in the mathematical sense.

Lower limit Instead of only evaluating the top-ranked hypothesis, we evaluate the average accuracy of all hypotheses, which corresponds to the lower limit in performance as it takes into account the success rate of less likely heatmaps as well. For $M = 5$, the accuracy for the shape-wise split drops to 75% from 85.3% (which corresponds to the top-ranked hypothesis). For a larger number of hypotheses, the accuracy drops further, which suggests noisier heatmaps overall. However, since the top-ranked score for $M = 10$ in Table 6.3 is higher than for $M = 5$, we believe that the Gaussian mixture fitting helps eliminate potentially bad hypotheses.

Upper limit Now we investigate whether the top-ranked hypothesis is indeed the best prediction in terms of grasp estimation accuracy. To compute the upper limit in accuracy we count grasp success is *any* of the hypotheses matches any of the ground truth rectangles, thus using an oracle for selection. For $M = 10$ the upper limit exceeds 99% accuracy for the shape-wise split, which implies that in almost all test samples there exists a successful grasp among the hypotheses, but it is not always selected, as the top-ranked hypotheses yield 86.2% accuracy for the same model. Therefore, there is room for improvement regarding the criteria for hypothesis selection.

6.4 Conclusion

We conclude the first part of the dissertation having addressed both scene-centric and object-centric problems. Object-centric understanding is as important as understanding scenes as a whole. An intelligent system deployed in the real world will need the ability to navigate and recognize its surroundings. However, it will be often also required to isolate target objects and, for example, identify points of interest to track the objects or potentially interact with them, especially when it comes to embodiment.

We have addressed this problem through two specific applications related to the field of robotic vision. In both applications, the problem we address is related to localization: a) localization of articulated surgical instruments (robotic or not) and b) localization of grasping points (*e.g.* as part of the visual processing system of a robotic gripper). In both applications, we cast the problem of localization into learning 2D heatmaps that represent points of interest on or around objects with some spatial uncertainty that accounts for ambiguous labeling. For example, in the first case precise labeling of surgical instrument joints is difficult and somewhat subjective. In the second case, annotating *all* viable grasping configurations around objects might not be done exhaustively. To address this second challenge in particular, we extend our model to predict multiple grasping hypotheses, thus allowing several options to the robotic arm, ranked by confidence. We show that this formulation significantly improves performance on this task.

Part III

Natural Language in Scene
Understanding

Image Captioning: Language as Output

7.1	Introduction	95
7.1.1	Motivation	95
7.1.2	Contribution	96
7.2	Related Work	98
7.3	Unsupervised Image Captioning	102
7.3.1	Language Model	103
7.3.2	Domain Alignment	106
7.4	Experiments and Results	111
7.4.1	Implementation Details	111
7.4.2	Unpaired Setting	113
7.4.3	Unsupervised Setting	118
7.4.4	Visualization of Embedding Space	119
7.5	Limitations and Discussion	121
7.6	Conclusion	123

7.1 Introduction

We have so far studied scene understanding tasks from a perceptual point of view. Often problems like recovering geometry and semantics are seen as lower-level tasks that enable higher-level tasks, such as navigation and interaction; ultimately, higher-level understanding and reasoning is the main goal towards intelligent agents. As lower-level visual perception problems are now being addressed by the research community with reasonable success, we have started witnessing the emergence of multi-disciplinary fields, such as the intersection of computer vision and natural language processing.

7.1.1 Motivation

There are several reasons why computer vision and image understanding are often coupled with natural language. First, a lot of information that is available on the web is in fact *multi-modal*; visual data are often associated with descriptions (*e.g.* Alt-Text) and images can be retrieved by search phrases or keywords. Other examples include books and their corresponding movies or movies with subtitles, social media photos and tags, and so on.

Second, priors, word co-occurrences, context or, in general, knowledge from human language can be exploited to guide visual tasks [332] or vice versa [174]. Indeed, humans also rely on compositionality and transferability to solve higher-level reasoning problems. Often knowledge for a task is transferred from other *modalities*, such as language — cooking a recipe after understanding the ingredients and instructions.

The goal of scene understanding is to teach machines to understand and reason about the world like human beings, which indeed involves multiple modalities. In this chapter, we motivate the important of *communicating* machine understanding, which is a higher-level, multi-disciplinary ability and is crucial in many practical scenarios that involve end-users, for example in aiding visually impaired users [143, 447], user-agent interaction [74, 254] or providing explanations and transparency into the system [176, 242].

We specifically focus on the task of automatically generating textual descriptions of images, which is also known as *image captioning*. This is an interesting problem because instead of restricting visual understanding to some limited-vocabulary representation (as in image classification or segmentation), it allows to extend understanding to how humans reason and talk about the world around them. Humans might look at a scene and very quickly be able to compose a detailed description, even at a young age — for example, children’s storybooks. On the other hand, automatically generating captions from images using a computational system is considered an extremely challenging task as it requires *joint and compositional understanding* on both image and language fronts. In other words, image captioning relies not only on an image model for scene understanding, but also on a language model that converts a scene representation into sentences that are correct both with respect to grammar rules and semantic visual entities (grounding). Due to these challenges, prior to deep learning and the availability of large image-caption datasets the amount of existing work on this task was limited [103, 216].

Image captioning has many potential applications in the real world. It is a crucial step to increase accessibility for blind people to computer applications, the web and social media or help them better understand their surroundings. There already exist wearable devices that help blind people read books, navigate streets and recognize other people around them by translating visual recognition into descriptions. Another example is the use of natural language to enable communication between the user and the system, which can help in gaining people’s trust and engagement when it comes to autonomous agents, such as self-driving cars. Last but not least, image descriptions can serve as a means for visual content search and image/video retrieval.

7.1.2 Contribution

The vast majority of existing methods on this task typically learns image captioning from full supervision, *i.e.* using large datasets of image-caption pairs annotated by human users on platforms such as Amazon Mechanical Turk (AMT). It is easy to notice the inherent ambiguity of such annotations, as there naturally exist many different ways to describe an image. For this reason, each image is usually annotated with more than one description — for ex-

ample, five in MSCOCO [253]. As turkers are usually paid per image, captions are quite repetitive and of low effort, resulting in datasets that need a manual process of cleaning and quality control. While it would be intuitive to mine the large amount of text already available on the web, this is still an open problem and, thus, most captioning models are developed on limited datasets alone that might be difficult to adopt in the real world.

More recently, researchers have started to move beyond curated datasets and explore various data sources and styles, by combining paired and unpaired sources. Following the potential that unsupervised methods have shown in other tasks [86, 230, 376, 490], we propose an approach to image captioning that does not require paired image and caption data. Unsupervised training can indeed benefit from an unlimited amount of unlabeled images (or also weakly labeled images, *e.g.* tags) and large text corpora (books, articles, descriptions, etc.). The absence of image-caption pairs gives rise to an interesting and challenging question: how can we align the two domains? This is the question that we investigate in this chapter, which has been published in [228].



KEY CONTRIBUTIONS

- We propose an unsupervised approach to image captioning, which reduces the dependency on paired image-caption data. At the time of publication, our method outperforms other unpaired methods.
- We propose a training scheme to build a language model with a *semantically structured* embedding space. In this space, sentences that describe similar visual content are encoded with similar embeddings.
- We thoroughly investigate several objectives for mapping images into the same embedding space with the goal that embeddings coming from different domains (image and language domain) are indistinguishable.
- Although visual recognition tools are necessary for establishing correspondences between the two domains, we show that, thanks to visual word co-occurrences, we can predict image descriptions that extend beyond the fixed vocabulary of an object detector.

It is important to note here that we classify our method as “unsupervised” *with respect to image-caption pairs*. However, when the image and language domains are independent, a weak supervisory signal — *e.g.* an off-the-shelf object detector — is in fact necessary for their alignment, *i.e.* we need to be able to *name* entities detected in the images. Therefore, a purely unsupervised approach in the image domain (*e.g.* unsupervised image classification [183, 416]) would not suffice. Nonetheless, as we show later, the co-occurrence statistics in the two domains help discover more concepts than the visual recognition tool that is used for the alignment.

Finally, our approach makes it possible to leverage various language styles (*e.g.* funny, poetic, romantic, narrative) or combinations of those. Fully supervised approaches, instead, would require obtaining such data through crowd-sourcing, which comes with its own challenges and limitations.

7.2 Related Work

The field of natural language generation from images has recently drawn a lot of attention. We will first review the literature related to image captioning and multimodal embeddings, which we divide as follows.

Pre-Deep Learning

Prior to the success of deep learning, several methods for generating image descriptions were based on templates [96, 103, 216], *i.e.* fixed template sentences with slots that were filled with objects, relationships or attributes detected in the image. While others address captioning as a retrieval and ranking problem [160, 311], some approaches are in fact compositional [222, 223, 245], combining together candidate descriptions retrieved from the training data.

While all these methods were impressive at the time, the generated language is much more rigid and constrained compared to the generative power that current neural networks allow.

Full Supervision

Some of the first approaches to use neural networks for image captioning were proposed by Vinyals et al. [423] and Karpathy and Fei-Fei [188]. The models proposed by these authors follow a similar concept: a CNN is used to encode an image into a representation which is then used to initialize an RNN for language generation. The problem is formulated as next-word classification by minimizing the cross entropy between the predicted and the real distributions. As the problem is auto-regressive in nature—and would require sampling a word at each step—one of the most common ways to train is with teacher forcing [36], *i.e.* predicting the next word in a sequence, given previous *ground truth* tokens. This is a well-established approach and several authors thereafter have used it as a basis for their methods.

Video captioning [87, 421, 467] is similar in nature, while the architecture design is adjusted to take temporal features into account. Also presented in [188], and later in [186], is a problem known as *dense captioning*, that is to caption several regions of an image individually instead of generating a single description for the entire image.

A substantial part of literature on image captioning has focused on attention mechanisms, following the seminal work of Xu et al. [459]. The purpose of attention is two-fold. First, it allows to put different weight on different parts of the image while generating each word (thus

regulating the focus of the decoder RNN), in a way that is implicitly learned. Second, it can be used as a means to visualize the grounding of generated words onto the image, thus improving the interpretability of such models. Several attention variants have been proposed since; for example, semantic attention [472], spatial *and* feature-wise attention [57], top-down and bottom-up attention [9], reviewer networks [466], visual sentinels that signify when (*i.e.* for which words) to attend [264] or the nowadays broadly used self-attention [418].

There is a significant amount of methods addressing various other aspects of image captioning. Wu et al. [445] and Yao et al. [468] use attributes and Yao et al. [469] detect visual relationships in the form of $\langle \text{subject}, \text{predicate}, \text{object} \rangle$ to create more explicit representations for the decoder.

When training with maximum likelihood estimation, the generated descriptions are usually rigid, unnatural and lack variability and the typically used metrics for evaluation do not penalize that. Thus, some approaches are based on Generative Adversarial Networks [71, 239, 374] or Variational Autoencoders [431] and aim to improve the naturalness and the diversity of the generated captions. However, GANs and VAEs are notoriously difficult to train due to the discreteness of the language domain and the auto-regressive nature of the decoder. To tackle this problem, and also to bridge the gap between training and testing modes¹, some methods [71, 343], inspired by reinforcement learning, propose to train using policy gradients [441]. Another advantage of these methods is that they can directly optimize for non-differentiable evaluation metrics, such as SPICE [12] or CIDEr [419].

Despite these noteworthy efforts, most methods have aimed at advancing the state of the art on common benchmarks such as MSCOCO which provide mostly *factual* descriptions of images. Thus, another line of work deals with stylized captioning to generate descriptions that can be more appealing to humans [114, 283, 284, 377]. SentiCap [284] proposes to use positive and negative sentiments and StyleNet [114] romantic or humorous style; Shuster et al. [377] condition on a large set of personality traits, such as *romantic*, *charming*, *arrogant*, etc. Notably, these methods rely on crowd-sourcing to annotate training data accordingly, while [283] follow a semi-supervised approach that we will discuss later.

Novel Object Captioning

The task of novel object captioning has been defined to address the shortcomings of captioning models in generating descriptions with previously unseen visual entities, *i.e.* that are not included in the paired image-caption training data. To achieve this, several methods exploit the fact that an abundance of visual categories are simultaneously present — although unpaired — in large-scale object recognition datasets (*e.g.* ImageNet [360]) and readily available text corpora. Their performance is usually evaluated on a held-out subset of MSCOCO.

Mao et al. [282] propose a few-shot learning approach to learn from just a few annotated pairs of images-captions that include novel concepts. Yao et al. [470] re-design the RNN architecture with a copying mechanism to exploit large-scale visual object recognition. Besides the paired captioning model, Hendricks et al. [17] train separate visual and linguistic

¹The gap arises from teacher forcing with ground truth during training and the decoder being exposed to its own predictions in auto-regressive testing (known as exposure bias).

models and learn a transfer function between categories that appear in the paired data and the novel (unpaired) categories, which also resembles a copying mechanism. Venugopalan et al. [420] improve upon this idea by employing joint, end-to-end training of all models. Lu et al. [265] revisit the older concept of template-based captioning (Baby Talk [216]) and fill in sentence slots with the outputs of an object detector. Finally, as an alternative to training with concepts that are not part of the paired data, Anderson et al. [10] modify beam search to constrain generation on a set of word targets.

The majority of these methods bear some similarity with our approach, as they try to leverage available imagery and large text sources independently, aside from paired datasets; however, the focus is on novel object categories, while the base captioning model is still fully supervised.

Partial Supervision

Most recently, a new and challenging avenue is being explored, which is closer to our work, that is to generate image captions in an unpaired or unsupervised manner. In this case, the *partial* supervision might refer to a number of things, such as adapting to different styles or languages without paired data.

Cross-domain captioning is the task of transferring a captioner trained on a source domain to a different domain, perhaps one in which paired data is not available. Chen et al. [59] address this task with an adversarial approach using two critics, one to account for the domain shift and one for the relevance of the description given an input image. Instead, Zhao et al. [483] formulate this problem with a cycle loss, *i.e.* reconstructing images from their descriptions. Instead of domain adaptation, unpaired text corpora of different styles are used in [283] to decouple captioning and stylization, where the styled text comes from romance novels. Interestingly, their method is not adversarial. Instead, [137] uses the already existing paired and stylized datasets, but in an unpaired way, and adopts an adversarial approach.

Anderson et al. [11] formulate the problem of image captioning as completing sequences of partially observed data — *e.g.* a set of object labels given by an object detector — using finite state automata, which is a more general approach to novel object captioning. This approach allows to perform captioning in an unpaired way, but is mainly shown in the context of describing new visual concepts. A different dimension of *unpaired* training is seen in [132], that captions images in a different language (without parallel data) by leveraging supervised machine translation [24] via a pivot language for which paired data do exist.

There is a crucial difference between the aforementioned methods and the approach that we will later describe in this chapter: in all previous cases, a base captioning model must be trained using parallel data and full supervision in a source domain, before it reaches its end-goal where there are limited or no paired data. Instead, our approach is unsupervised in the sense that it does not entail a dependency on a paired image-language domain; concurrent work [106, 133] has a similar objective. In this case, we can make use of large text corpora with the only constraint being that they are sufficiently descriptive in terms of visual concepts, thus suitable for describing visual content. Similar to our method, Feng et al. [106] only make use of an object detector to recognize visual entities in the images and create a set

of pseudo-captions for training. On the other hand, Gu et al. [133] use scene graphs, which is a much stronger supervisory signal, as it includes enough information (objects, relationships and attributes) to induce accurate descriptions even without aligned data.²

Multimodal Embeddings

A significant contribution of our approach is the translation of visual and linguistic representations into a common multimodal space, without using parallel data. Thus, we also discuss existing work related to multimodal embedding spaces.

In particular related to image captioning, there exist methods that operate on a multimodal space. Instead of directly decoding a visual representation into captions, multimodal methods create a shared latent space for images and captions and decode features from this space. In pioneering work, Kiros et al. [204] extend log-bilinear models (originally proposed in [297] for language modeling) to learn a joint embedding space, which can be then used for either image or caption retrieval as well as caption generation. In follow-up work [206], they propose an encoder-decoder approach, where a sentence encoder is modeled by an LSTM and the decoder is a neural language model that predicts both sentence structure and contents and can be trained using text corpora only. Karpathy et al. [189] break down images and sentences into “fragments” via an object detector and dependency trees [381] respectively and learn multimodal embeddings from both global and fragmented levels. Fang et al. [102] generate candidate sentences by predicting common visual words in images and use a multimodal similarity model to rank the candidates. Faghri et al. [101] propose an improvement to the standard ranking loss for multimodal embeddings by considering hard negatives. Focusing on word level, DeViSE [110] is a visual-semantic embedding model that is trained by using the last layers of an image encoder to predict word vector representations with a ranking objective. Different to all other approaches, Chen and Zitnick [62] propose a bi-directional mapping by additionally predicting visual features from text data. Similarly, multimodal embeddings have been also used for related problems, such as video captioning [314], book-movie alignment [491] and visual question answering [200].

More recently, in the field of unsupervised machine translation, although dealing with linguistic representations only (unimodal), [229, 230] learn an aligned latent space from non-parallel monolingual corpora with a reconstruction objective in both languages. As we show later, in (unsupervised) image captioning, we have two modalities with quite different properties; thus, to improve their alignment, we create representations in a way that is specifically directed by *visual* cues.

Sentence Representations and Topic Modeling

Another key component of our method is creating sentence representations that are (mostly) unaffected by syntactic structures and encode visual information instead. This topic has been previously studied in [381] using dependency tree relations and constructing sentence representations by operating on word representations in a bottom-up approach.

²The scene graphs have been created from image descriptions in the first place.

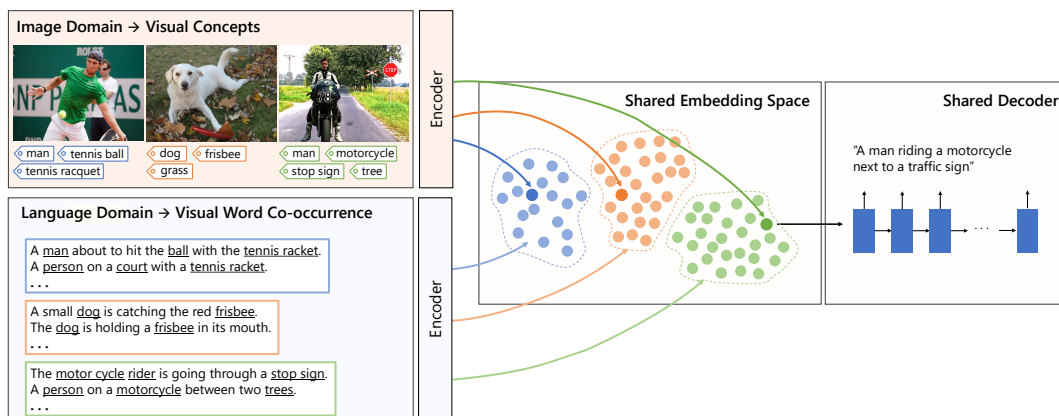


Figure 7.1 Method overview. We learn a joint embedding space of text and image features from disjoint language and image domains. The embedding space is structured by visual concepts and their co-occurrence. In the language domain, visual words are underlined. A shared decoder can decode image and text embeddings in the same manner and thus unsupervised captioning can be achieved when the two domains are aligned.

Sequence-to-sequence models are a well-established approach to encoding (and decoding) sentences [390]. There exist several other methods that deal with unsupervised learning of sentence representations [156, 205] and topic modeling [40]. These methods are not related to the visual domain.

7.3 Unsupervised Image Captioning

In the following, we present our approach to unsupervised image captioning. The key idea is to learn a *shared embedding space* into which we can map vectors from the visual domain and text corpora *independently*. In this space, image representations and sentence representations should be indistinguishable; this effectively means that the decoder of a language model — pre-trained on the text corpus — can decode embedded images just as if they were embedded sentences instead. Thus, as a broad overview our model consists of two parts; the language model and the alignment between the two domains in the embedding space, *without* image-caption correspondences. This overview is also illustrated in Figure 7.1.

In Section 7.3.1 we present our language model and elaborate on why imposing a semantic structure is more suitable for the task at hand. In Section 7.3.2 we first create initial noisy correspondences between the two domains that will then act as pseudo-supervision and then present the full model and training objectives that are used to translate visual features to textual representations, thus bringing *multi-modality* into the embedding space.

We will first introduce some general concepts and notation that will be used throughout the chapter. Let \mathcal{J} be the visual domain that contains images $I_i \in \mathcal{J}$, where i is used to index image samples. We also represent each image by the set of visual entities depicted in it:

$$\mathcal{V}_i = \{v_k \mid k \in \mathbb{N}, 1 \leq k \leq N_i\}, \tag{7.1}$$

where N_i is the total number of visual entities. Within the context of our work, visual entities mostly refer to object categories, but can be extended without loss of generality to object interactions (predicates), attributes, etc.

Independently, there exists a language domain (text corpus) \mathcal{C} and sentences $s_j \in \mathcal{C}$, where j is used to index sentence samples. We choose a similar representation for sentences as we did for images, *i.e.* we use a set \mathcal{W}_j of “visual” words:

$$\mathcal{W}_j = \{w_k \mid k \in \mathbb{N}, 1 \leq k \leq M_j\}, \quad (7.2)$$

where M_j is the number of relevant words in sentence j . It is important to note that not all words in \mathcal{W}_j are actually needed to represent the sentence, for example we disregard prepositions, articles, etc.



NOTE: *Visual words*

In computer vision, the term *bag of visual words* has been used to represent images as collections of features (*e.g.* SIFT [261]), in particular for tasks such as image classification. In the context of this dissertation, we refer to *visual* words with a literal meaning, *i.e.* words in our text corpus that have a visual meaning and can be explicitly grounded in imagery (as in [265]).

Here we are specifically interested in visual nouns. We extract these from the sentences using the Stanford part-of-speech tagger [281] and a lookup table provided with the Visual Genome dataset [214]. In practice, this lookup process results in *synsets* (WordNet [292]), which we prefer over raw words so that synonyms, plural forms, etc. still refer to the same entities (*e.g.* `bike` and `bicycle` are the same concept).

Although our method does not depend on paired data, we must make the assumption that the image domain and language domain are somewhat related. This is simply because we cannot expect to generate image *descriptions* if we use, for example, text corpora of legal matters or political speeches. Therefore, we select the language domain(s) such that there exists a set of *visual* concepts that is shared between images and text corpora, *i.e.* $\Omega = \mathcal{V} \cap \mathcal{W}$ and $\Omega \neq \emptyset$.

7.3.1 Language Model

The first step of our approach exclusively involves the language domain, that is to pre-train a language model and learn meaningful sentence representations that will form the basis for the subsequent alignment of the image and language domains.

Sequence-to-Sequence Models

We build on sequence-to-sequence models [390] and train encoder-decoder RNNs with maximum likelihood estimation. This is the standard approach to sequence learning that maps an input sentence s to a d -dimensional latent vector using an encoder $f(\cdot)$ and then decodes

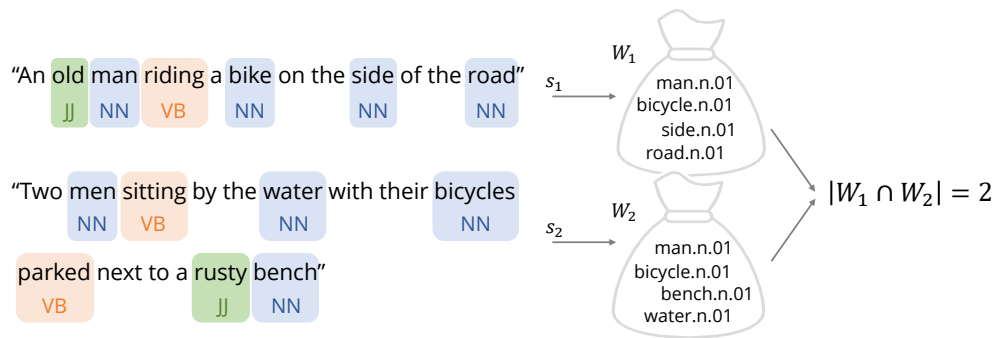


Figure 7.2 Bags of visual words. We show two sample sentences and simplified part-of-speech tagging. We extract synsets from the detected visual nouns to create bags W_1, W_2 and use those to denote the two sentences as positive to each other since they have two synsets in common (`man.n.01` and `bicycle.n.01`).

this vector into a target sentence using decoder $g(\cdot)$. In our case, the target sentence is the same as the input sentence, *i.e.* we formulate the training task as sentence reconstruction as a means for unsupervised representation learning.

$$f(s) = \phi, \quad g(\phi) = \tilde{s}, \quad \phi \in \Phi \subseteq \mathbb{R}^d. \quad (7.3)$$

The most common choices for f and g are LSTMs [159] or GRUs [66]; we choose the latter. The goal is to compute the conditional probability of the output

$$p(\tilde{s} | \phi) = \prod_{t=1}^T p(\tilde{s}^t | \phi, \tilde{s}^1, \dots, \tilde{s}^{t-1}), \quad (7.4)$$

where t is used for indexing sentence tokens and $p(\tilde{s}^t)$ is a distribution over words from a fixed-size vocabulary. The model is trained to minimize the negative log-likelihood of a correct reconstruction of the input sentence:

$$\mathcal{L}_{\text{CE}}(s, \tilde{s}) = - \sum_{t=1}^T \log p(\tilde{s}^t = s^t | s^1, \dots, s^{t-1}) \quad (7.5)$$

Since Equation (7.4) implies an autoregressive nature for the decoder — each output token is conditioned on the previous ones — it also involves non-differentiable token sampling. As we discussed previously, to address this, language models are trained with teacher forcing, *i.e.* given *ground truth* words s^1, \dots, s^{t-1} as input at each time step, the goal is to predict the next word; which brings us to Equation (7.5).

Visually Structured Language Model

Given the nature of the task we are dealing with, here we are specifically interested in learning representations that encode (visual) semantics. Instead, we have observed that the model described above learns embeddings that are sensitive to grammatical and syntactic structures, such as active/passive voice. Our observation aligns with the findings of

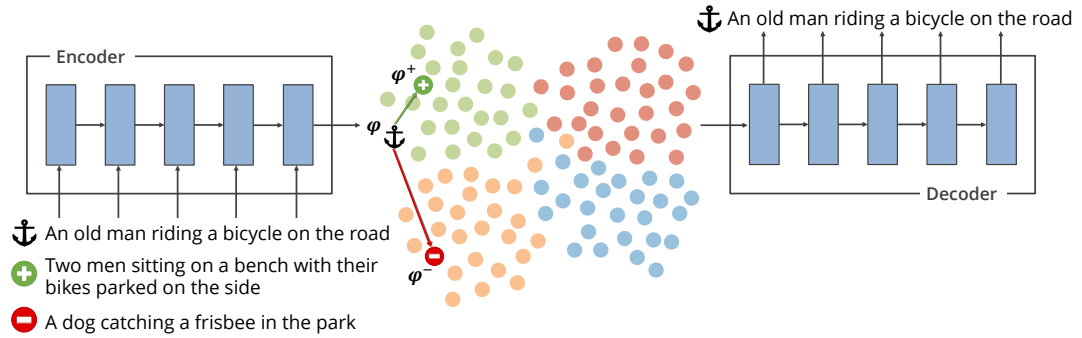


Figure 7.3 Language model. We build an encoder-decoder language model which, in addition to sentence reconstruction, is trained with sentence triplets. The triplet loss creates an embedding space that is structured by the visual concepts that are present in the language domain.

[381]. While this structure might be important for other tasks, we argue that for image captioning it would be optimal if sentences with similar visual representations also have similar embeddings, *i.e.* a small distance in the embedding space. Therefore, our goal is to learn a language model that can recognize words with *visual grounding* and encode abstract visual concepts that stem from the *co-occurrence* of these words.

To achieve this, we create sentence (and therefore embedding) triplets (ϕ, ϕ^+, ϕ^-) , where ϕ^+ corresponds to a sentence with similar visual content as ϕ , while ϕ^- has dissimilar content. Visual awareness can emerge in the embedding space by minimizing the triplet loss

$$\mathcal{L}_t(\phi, \phi^+, \phi^-) = \max(0, \|\phi - \phi^+\|_2^2 - \|\phi - \phi^-\|_2^2 + m). \quad (7.6)$$

The triplet loss is minimized when the distance between the anchor embedding ϕ and its positive pair ϕ^+ is lower than the distance between ϕ and the negative pair ϕ^- by at least a margin $m \in \mathbb{R}^+$.

Since the positive and negative pairs depend on the similarity of visual contents among sentences, it naturally follows that for a sentence s_j the set \mathcal{S}_j^+ of positive candidates can be defined as the sentences that have at least some visual words in common with s_j :

$$\mathcal{S}_j^+ = \{s_k \mid k \in \mathbb{N}, k \neq j, |\mathcal{W}_k \cap \mathcal{W}_j| \geq 2\}. \quad (7.7)$$

On the other hand, the set \mathcal{S}_j^- of negative candidates can be defined as the sentences that have nothing in common with s_j :

$$\mathcal{S}_j^- = \{s_k \mid k \in \mathbb{N}, \mathcal{W}_k \cap \mathcal{W}_j = \emptyset\}. \quad (7.8)$$

According to Equation (7.7) we have defined sentences as similar when the number of visual words in common is at least two. A single word overlap is not enough to define similarity, since it cannot be used to model more complex scenes. For example, all sentences that

contain the word *person* or a synonym would be a positive to each other disregarding all context. We illustrate this process with an example in Figure 7.2.



NOTE: *Example*

Given the above definition, the sentence “Two men sitting by the water with their bicycles parked next to a rusty bench” is positive to the sentence “An old man riding a bike on the side of the road” and also to “A bench surrounded by water” due to the underlined visual nouns, but it is negative to “A dog running after a frisbee in the park”.

In the previous example, we see that a sentence can be assigned positives that depict substantially different scenes, which allows us to learn more abstract concepts from visual word compositions and partial overlap.

The total objective we use to train the language model becomes

$$\mathcal{L}_{\text{LM}}(s_j) = \mathcal{L}_{\text{CE}}(g(\phi), s_j) + \lambda_t \mathcal{L}_t(\phi_j, \phi_j^+, \phi_j^-), \quad (7.9)$$

thus the cross-entropy loss operates on the output space while the triplet loss operates on the latent space of the language model and its contribution is weighted by factor λ_t . For a given training sample s_j , a positive sentence $s_j^+ \in \mathcal{S}_j^+$ is sampled from a multinomial distribution at each iteration of training with probability proportional to the number of common concepts (such that sentences with more synsets in common are more likely to be selected as positive). From s_j^+ the same encoder extracts ϕ_j^+ . Negative sentences $s_j^- \in \mathcal{S}_j^-$ are sampled uniformly and features ϕ_j^- are extracted.

Our language model is shown in Figure 7.3. We further show the effect of the triplet loss on the manifold of sentence embeddings in Figure 7.9, by comparing the visually unaware (trained with Equation (7.5)) and visually aware (trained with Equation (7.9)) language models. As image captioning is related to understanding and describing visual content, we expect (and experimentally verify in Section 7.4) that a visually structured embedding space is better suited for this task.

7.3.2 Domain Alignment

We have so far learned a language model, the encoder of which is able to map sentences into a semantically structured embedding space and the decoder translates these embeddings into sentences. However, since our end-goal is image captioning, we need a way to decode *image vectors* into sentences instead. The key idea is to translate images into the same embedding space and make them indistinguishable from the sentence vectors, such that they can be decoded interchangeably using the same decoder. We do so by creating pseudo-correspondences between the image and language domains based on detected vi-

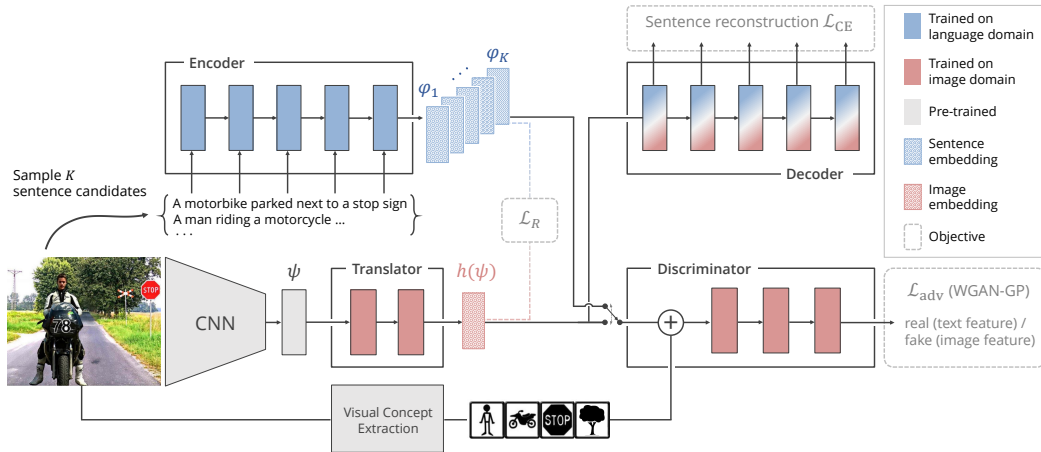


Figure 7.4 Domain alignment model. For each image we sample a set of candidate sentences, which are encoded using the previously trained language model. The image features are projected into the same embedding space, minimizing a robust loss towards the sentence embeddings, a conditional adversarial loss and the sentence reconstruction loss.

sual concepts and using these correspondences as training pairs to supervise the full model for domain alignment.

Cross-Domain Correspondences

To create initial correspondences between the two domains, we build a bipartite graph $\mathcal{G}(\mathcal{C}, \mathcal{J}, \mathcal{P})$ with images I_i and sentences s_j as nodes and edges $P_{i,j}$ that denote a *weak* pairing between I_i and s_j . The edges are weighted by the number of visual concepts that images and connected sentences have in common, *i.e.*

$$P_{i,j} = |\mathcal{V}_i \cap \mathcal{W}_j|, \quad (7.10)$$

which are determined based on Equation (7.1) for images and Equation (7.2) for sentences. When training for domain alignment, we use these pairs as pseudo-supervision and sample for each training image I_i a sentence s_j with probability

$$p(s_j | I_i) = \frac{P_{i,j}}{\sum_k P_{i,k}}. \quad (7.11)$$

We note that for training we only consider pairs with $p(s_j | I_i) > 0$, while those that have nothing in common are excluded from training. It also follows that candidate pairs with many visual concepts in common will be sampled with higher probability.

We have already discussed how bags of visual concepts \mathcal{W}_j are extracted in the language domain. In the image domain, visual entities are extracted using any pre-trained image recognition method — which can be, for example, either multi-class, multi-label classification [151, 378] or object detection [342]. Crucially, using recognition networks trained on datasets such as ImageNet [360] or OpenImages [221] allows to identify a large number of

visual categories that can boost the quality of the pseudo-assignments. However, directly searching for sentences that include specific object category names (e.g. “person”) produces only limited matches. Therefore, we expand the search using lexical relations from the WordNet ontology [292], specifically synsets (synonyms) and hyponyms (subordinates) of the predicted object categories. For example, when *person* is detected in an image, potential sentence candidates (or their bags \mathcal{W}_j) may contain the synset `person.n.01`, but also possibly `man.n.01`, `woman.n.01`, `child.n.01`, and so on.

The use of a pre-trained image recognition network is the only form of supervision we use in our approach and is necessary for naming entities in images and therefore for translating the visual world to words.

Learning to Align Image and Text Embeddings

Having built correspondences between the image and language domains allows us to learn a mapping from images to sentences (descriptions). The simplest way to do so would be in the fashion of fully supervised methods such as [423] with the only difference being that our dataset $(I_i, s_j), \forall i, j$ is stochastic and we sample a different sentence as the target at each training step. It is to be expected that a lot of the sentence candidates will only have a weak to almost no relation to the actual image content, which is a bad supervisory signal. In fact, the more detailed a sentence is, the less fitting it might be for a given image. Surprisingly though, this approach forms a strong baseline. The model mainly learns the co-occurrence of visual concepts and eventually eliminates noisy context. On the other hand, this behavior makes it prone to simpler sentences that are repeatedly generated for many similar images.

We propose an improvement over this naïve approach by building a *multi-modal* embedding space where image and text features co-exist, instead of directly decoding image representations. The basis of this shared embedding space is the latent space created after training the encoder-decoder language model in Section 7.3.1.

Next, let $\psi_i \in \Psi \subseteq \mathbb{R}^c$ be an image vector extracted from I_i using a standard feature extractor (such as a CNN), with c denoting the vector dimensions. Then, our goal is to map this vector into the embedding space Φ using a translation function $h : \Psi \rightarrow \Phi$, which can be modeled as a multi-layer perception (MLP).

The full model can be seen in Figure 7.4, though modules that appear as blue or gray are not updated in this step. To learn the mapping from images to shared embeddings using this setup, we also propose the following training objectives.

Robust Alignment A straightforward way to align image and sentences embeddings would be to minimize a notion of distance, for example using

$$\mathcal{L}_2(\psi_i) = \sum_j \|\mathbf{h}(\psi_i) - \phi_j\|_2^2, \quad (7.12)$$

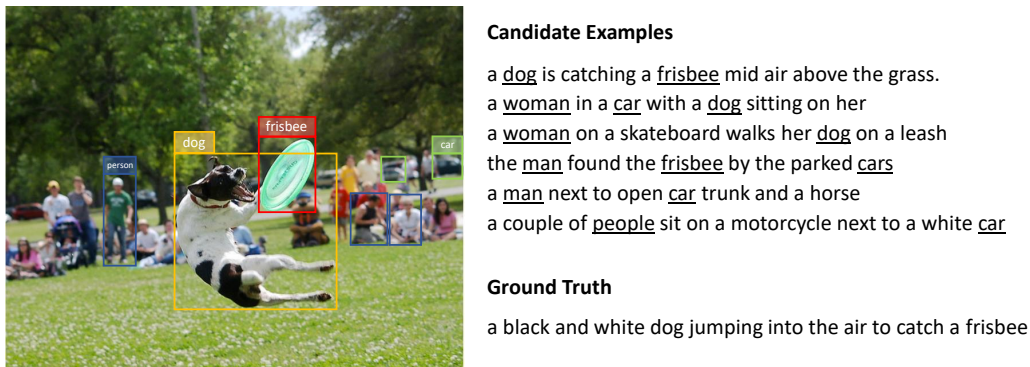


Figure 7.5 Example on image-caption correspondences. The image is overlaid with detected object categories from MSCOCO. The candidate sentences are real annotations from MSCOCO. Although all of them have at least two entities in common with the image, they exhibit very different degrees of correlation.

treating embeddings ϕ_j as ground truth and not back-propagating through the language model. Equation (7.12) aims to minimize the distance between a single image embedding and *all* pseudo-assigned sentences. Then, the optimal solution for h would be the conditional average $h^*(\psi_i) = \sum_j p(\phi_j | I_i) \phi_j$.

There is an intuitive explanation as to why this is actually far from ideal, which we also illustrate with an example in Figure 7.5. Due to the assignments being rather approximate and solely dependent on visual entities with no notion of saliency, *i.e.* which entities in the image are actually relevant to describe, candidate feature vectors ϕ_j might be scattered all over the embedding space. For example, for the image shown in Figure 7.5, we observe different degrees of relevancy to ground truth MSCOCO captions, with “a dog is catching a frisbee mid air above the grass” accurately describing the image and “a couple of people sit on a motorcycle next to a white car” being entirely out of context, despite having the same probability to be sampled during training. In this example, the latter sentence is perhaps part of a cluster related to traffic scenes, while the former belongs to a cluster of sentences that might be overall a preferable match for this image. Learning the conditional average h^* can be detrimental when mapping images into the embedding space as it is directly affected by candidates that lie in different “visual clusters”, potentially pushing the translated image feature away from better candidates.

To tackle this issue, we propose a robust formulation for the alignment objective:

$$\mathcal{L}_R(\psi_i) = \min_{\phi_j \sim p(s_j | I_i)} \|h(\psi_i) - \phi_j\|_2^2. \quad (7.13)$$

The minimum formulation encourages the image features to land near the sentence embedding that is already the closest (in Euclidean distance) in the embedding space. In practice, since the number of sentence candidates for a given image is usually quite high, we only sample a sub-batch of K sentences at each iteration of training, based on their corresponding probabilities. Then we approximate Equation (7.13) by computing the minimum between $h(\psi_i)$ among the K feature vectors. Only the minimum-distance sentences contribute to each iteration of training, which is a more robust criterion.

Adversarial Training To further encourage the alignment of the translated image vectors with the existing sentence embeddings, we propose to use adversarial training. Training with adversaries for image captioning would typically require sampling an entire sentence at the output of the generator and trying to fool a discriminator into believing this is a real caption for a given image [71]. It is well known that this is a non-trivial problem due to the instabilities of both minimax adversarial training and the non-differentiable sampling which needs to be addressed with reinforcement learning [392, 441]; thus this approach is usually combined with standard maximum-likelihood training.

Instead, we follow the example of [388] and perform adversarial training in feature space, which significantly increases stability during training. Effectively this means that the discriminator is trained with a set of real and fake *features* as input, where the real features come from the language domain (sentence embeddings ϕ) and the fake features come from the image domain (translated image embeddings $h(\psi)$). The discriminator is optimized for accurately classifying these, which pushes the translator h into producing more and more realistic embeddings.

In fact, we observed that this is not sufficient and the discriminator can easily tell apart real sentence embeddings from translated ones. In this case the goal of the discriminator would be equivalent to identifying real sentences in the discrete case, unconditionally. Naturally, sentence generation is conditioned on images when it comes to image captioning. In a similar fashion, we choose to condition the feature discriminator on visual concepts to learn not only “realistic” features but also the kind of image content that they could possibly describe. We condition the discriminator by concatenating a binary vector of the visual concepts \mathcal{V}_i detected in the image to the feature vectors. This conditioning then encourages the translator to better encode image concepts in its output.

We use the formulation of Wasserstein GAN with gradient penalty (WGAN-GP) [134]. With this formulation the discriminator $D : \Phi \times \Omega \rightarrow \mathbb{R}$ has the role of a *critic*, *i.e.* it outputs a score that denotes *how real* the input is instead of a probability. D can be implemented as a simple MLP. The critic loss is:

$$\mathcal{L}_{\text{crit}} = \mathbb{E}_{\psi, \mathcal{V} \sim \mathbb{P}_{\psi}} [D(h(\psi), \mathcal{V})] - \mathbb{E}_{\phi, \mathcal{V} \sim \mathbb{P}_{\phi}} [D(\phi, \mathcal{V})] + \lambda_{\text{gp}} \mathbb{E}_{\tilde{\phi}, \mathcal{V} \sim \mathbb{P}_{\tilde{\phi}}} [(\|\nabla_{\tilde{\phi}} D(\tilde{\phi}, \mathcal{V})\| - 1)^2], \quad (7.14)$$

where $\tilde{\phi}$ are random linear interpolations between text features and translated (image) features: $\tilde{\phi} = \epsilon\phi + (1 - \epsilon) * h(\psi)$ and ϵ is a random number between 0 and 1. λ_{gp} controls the strength of this *gradient penalty* term.

Finally, h which plays the role of the generator is trained with:

$$\mathcal{L}_{\text{adv}} = -D(h(\psi_i), \mathcal{V}_i), \quad (7.15)$$

i.e. maximizing the discriminator’s score for mapping the image features into the embedding space.

Maximum Likelihood Estimation The previous objectives aim at creating a multi-modal space, where image and text features are indistinguishable. Nevertheless, since the alignment cannot be perfect, it is necessary to update the decoder g such that it co-adapts to the new embedding space. We do so by finetuning the pre-trained decoder with maximum likelihood estimation, *i.e.* using \mathcal{L}_{CE} against the ground truth sentence that is the minimizer of Equation (7.13) for each image at each training iteration. The initial hidden state of the decoder is now set to $h(\psi)$ instead of ϕ .

Total Objective The full model, which can be seen in Figure 7.4, is trained with all the above objectives:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{CE}} \mathcal{L}_{\text{CE}} + \lambda_{\text{R}} \mathcal{L}_{\text{R}} + \lambda_{\text{adv}} \mathcal{L}_{\text{adv}}, \quad (7.16)$$

where $\lambda_{\text{CE}}, \lambda_{\text{R}}, \lambda_{\text{adv}} \in \mathbb{R}$ balance the contribution of each loss term.

7.4 Experiments and Results

We have conducted experiments under two settings, the *unpaired* MSCOCO setting and an *unsupervised* setting. When using the *unpaired* setting we make use of images *and* captions from MSCOCO, however we do not consider any correspondences between the two. We use this setting to perform ablation experiments and fair comparisons to previous work. In the *unsupervised* setting we make cross-dataset experiments using different text corpora, such as captions from the Conceptual Captions dataset (GCC) [371], questions from VQA-v2 [19] or book narratives.

7.4.1 Implementation Details

Before evaluating our method and reporting results, we give important implementation details.

Sentence Pre-Processing

We tokenize all sentences and add special tokens $\langle \text{start} \rangle$ and $\langle \text{end} \rangle$ to denote the beginning and end of each sentence. Infrequently occurring words are replaced with the $\langle \text{unk} \rangle$ token. We set the frequency threshold to 50 for MSCOCO and GCC.

At the same time, the bag representations are created by parsing each sentence with the Stanford CoreNLP toolkit [281], extracting the nouns and then converting them to synsets using the lookup table from Visual Genome [214]. This process results in a vocabulary comprised of *visual* synsets (for example, 1415 for MSCOCO and 3030 for GCC). We use these to define positive and negative pairs of sentences when pre-training the language model with the triplet loss.

**NOTE:** *Visual Relationships*

It is possible to augment the model with visual relationships that define interactions between visual entities (such as *wearing*, *riding*, *below*, etc.). These are usually not nouns, thus one could also consider verbs or prepositions when parsing the sentences and add them to the visual synset vocabulary.

Language Model

We use GRUs [66] for the implementation of the encoder and decoder of the language model, with 200 hidden units each. Input to the encoder are 200-dimensional word embeddings from the GloVe implementation [317]. A linear layer maps the last hidden state of the encoder to a 256-dimensional feature vector ϕ . This then becomes the initial hidden state of the decoder. Another linear layer followed by softmax maps the output of the decoder into a probability distribution over all words in the vocabulary (3463-d for COCO and 14305-d for GCC).

The language model is trained from scratch on the selected text corpus with a batch size of 64 for 70 epochs using Adam optimization [202]. The initial learning rates are set to 10^{-4} for the encoder and 10^{-3} for the decoder. The weight factor for the triplet loss is $\lambda_t = 0.1$ and the margin $m = 0.2$.

Visual Concept Extraction

To extract visual concepts from images we use Faster R-CNN [173, 342] trained on Open-Images v4 [221], which consists of 1.74 million images and includes 600 object categories. This model is pre-trained and publicly available.³ This dataset allows for much more fine-grained recognition than detectors trained on MSCOCO, which includes just 80 classes. The choice of the detector is crucial as it defines all the identifiable entities and therefore has an impact on the quality of the initial assignments between the image and language domains. However, we only need class labels and do not actually make use of the bounding box coordinates. This is a choice we have made in order to minimize the annotation effort that is actually required overall in our framework.

Domain Alignment Model

The (fully supervised) baseline for our image captioning approach is Show-and-Tell [423], with ResNet-101 [151] acting as the image encoder. This is equivalent to our model without the multi-modal embedding space. The sentence encoder and decoder come from the previously trained language model. The encoder weights are not updated; it now only serves as a feature extractor. The architecture for the translator h is an MLP with two hidden layers of 512 units each that project input features $\psi \in \mathbb{R}^{2048}$ to $\phi \in \mathbb{R}^{256}$. The discriminator is an MLP with a single hidden layer of 200 units followed by leaky ReLU.

³https://github.com/tensorflow/models/tree/master/research/object_detection

Before training the full model for domain alignment, we pre-train the image encoder on MSCOCO with sigmoid cross-entropy for multi-label classification. We then freeze the encoder and use adaptive average pooling on the last layer to extract a 2048-dimensional feature vectors from images. We then jointly train the translator, discriminator and decoder using the aforementioned objectives with $\lambda_{CE} = \lambda_R = 1$, $\lambda_{adv} = 0.1$ and $K = 10$ (unless otherwise noted). We use Adam with a learning rate of 10^{-4} for the translator and decoder and 10^{-5} for the discriminator and a batch size of 64. We train for up to 150 epochs but choose the epoch that gives the best validation performance.

Evaluation Metrics

To evaluate our method we use the official COCO evaluation code. In the following experiments we report our method’s performance under commonly used metrics: BLEU 1-4 (B1-4) [66], ROUGE (R) [250], METEOR (M) [80], CIDEr (C) [419], SPICE (S) [12] and WMD (W) [220].

7.4.2 Unpaired Setting

We first experiment with unpaired images and captions from MSCOCO to evaluate the contribution of each proposed component in our approach and to compare to the state of the art on unsupervised image captioning. We follow the split suggested by Karpathy and Fei-Fei [188], which results in 113,287 training, 5,000 validation and 5,000 test images. Every image in MSCOCO is annotated with 5 captions, which results in over 560k training captions in total.

Ablation Study

Ablation experiments are reported in Table 7.1 evaluating all proposed components. In these experiments, we use the *ground truth* MSCOCO object categories as visual concepts in the image domain. We do so to eliminate any influence of object detector errors on the ablation. Since our dataset preparation relies on matching images and captions via visual synsets, only 150k unique captions remain in our language domain (the rest contain words that were not matched to MSCOCO categories).

We compare and discuss the following variants.

Supervised The supervised baseline builds on the model of [423]. We use the same specifications for the image encoder and language decoder as in the rest of our experiments. The model is trained on the full, paired MSCOCO dataset.

Oracle To get an idea on the performance of the weak image-caption assignments of the graph \mathcal{G} we build as supervision, we evaluate an oracle that selects the most likely candidate caption for each image, *i.e.* the caption that has the most visual concepts in common with the image. However, by construction, there usually exist many captions with equally high probability for an image, thus we randomly sample among them and report the results

Component Evaluation					Metrics								
Abbreviation	\mathcal{L}_{CE}	\mathcal{L}_2	\mathcal{L}_R	\mathcal{L}_{adv}	B1	B2	B3	B4	M	R	C	S	W
Supervised					67.4	50.0	35.4	24.8	22.6	50.1	80.2	15.9	17.9
Oracle					49.1	31.2	21.2	16.0	18.7	38.7	50.4	12.2	14.5
MLE only	✓				59.9	40.2	26.0	17.1	19.1	43.7	57.9	11.6	13.0
Alignment only													
baseline		✓			47.0	25.4	11.5	5.2	15.5	35.9	29.4	8.7	9.1
robust			✓		53.1	31.8	17.9	10.5	17.5	40.1	42.3	10.3	11.3
adversarial			✓	✓	55.1	33.2	19.0	11.2	17.1	39.8	42.7	10.4	11.5
Joint													
baseline	✓	✓			59.7	40.2	25.8	16.6	18.3	43.1	53.8	10.8	12.6
robust	✓		✓		61.0	42.2	28.1	19.0	19.4	44.7	61.3	12.3	13.9
robust ($\lambda_t=0$)	✓		✓		60.7	41.1	26.7	17.6	18.3	43.8	55.6	11.0	13.0
adversarial	✓		✓	✓	61.7	42.8	28.6	19.3	20.1	45.4	63.6	12.8	14.4

Table 7.1 Ablation experiments on MSCOCO test set [188]. We do not use pairs of images and captions when training. MSCOCO ground truth object categories are used as visual concepts in the image domain. All proposed components positively add to the performance on the captioning task.

for the best out of 100 runs. Although this baseline directly chooses among ground truth captions, all metrics except for WMD are lower than those of our trained model because the initial assignments are actually very noisy. This experiment further suggests that our model is able to learn concepts beyond the initial weak assignments and filter out the irrelevant ones.

MLE only We train the model in a “supervised” fashion, with cross entropy as the objective function (and teacher forcing), but we sample our training data from the weak image-caption assignments instead of using the real pairs. In this model, there is no constraint enforcing a multi-modal embedding space. The model is prone to biases often observed in MLE models, for example limited novel captions or repeating sub-phrases. Nevertheless, it is a surprisingly strong baseline.

Alignment only In this set of experiments we train only the translator h to learn the mapping of images into the embedding space created by Φ , thus forming a multi-modal space. We keep the decoder weights fixed after pre-training the language model, *i.e.* we do not finetune the decoder with image features. We compare the baseline that simply minimizes the \mathcal{L}_2 distance between features to our robust (\mathcal{L}_R) and adversarial ($\mathcal{L}_R + \mathcal{L}_{adv}$) objectives, which do indeed bring a considerable improvement over the baseline. Overall, we observe that the alignment is successful in that major visual concepts from the images are described in the output sentences. However, the performance remains worse than MLE training for all metrics. This is because the decoder cannot adapt to the differences between the translated image features and the real sentence features it had seen during training, producing captions that are grammatically and syntactically incoherent. To better illustrate this behavior, we also show some qualitative examples in Figure 7.6.

For the following experiments we found that it is necessary to jointly train the decoder and domain alignment module (thus we refer to this as **Joint** training).

Joint, baseline In addition to training with MLE, we create a multi-modal feature space by minimizing the \mathcal{L}_2 distance between $h(\psi)$ and ϕ . However, this naïve approach to domain alignment does not actually improve over the MLE-only baseline.

Joint, robust We then train the model with the proposed robust loss (\mathcal{L}_R) and $K = 10$ samples. This adds several points of improvement in performance under all metrics.

Joint, robust ($\lambda_t = 0$) At this point, we also evaluate the importance of a visually structured embedding space Φ by learning the alignment using sentence embeddings from a language model that was trained *without* the triplet loss ($\mathcal{L}_{LM} := \mathcal{L}_{CE}$). Comparing to the previous experiment, this results in worse performance, verifying our intuition that visual-semantic awareness in the embedding space is helpful for this problem.

Joint, adversarial Finally, we present our model trained with all three objectives, which achieves the best performance with a significant margin. In fact, this model reaches performance close to our fully supervised variant and early methods on image captioning.

Comparison to Previous Work

Image captioning without image-caption pairs has been only addressed very recently in literature. We compare our approach to existing methods in Table 7.2 under the same unpaired setting as [106]. To extract visual concepts from the images and draw correspondences between images and captions we use the detector trained on OpenImages (OID). The predicted object categories are also used to condition the discriminator during training. The detector is not used during the testing phase, unless otherwise state (*e.g.* in targeted beam search). To fairly compare our method, we generate captions using beam search with a beam size of 3. We have included the method of Gu et al. [132] for completeness although it is not directly comparable to ours, as the authors deal with a different unpaired problem (using an intermediary language). On the other hand Feng et al. [106] address exactly the same problem. Our method outperforms prior work.

NOTE: *Beam Search*

During *greedy* sampling, we sample the highest probability word from the output of the decoder at every step of the generation. When the $\langle \text{end} \rangle$ token is sampled, it signifies the completion of the sentence. *Beam search* [209] is a heuristic method that allows to rank multiple sentences according to their likelihood. Instead of greedily choosing the most probable word, beam search considers the k most likely next words for each candidate sentence and selects the k most likely candidates overall before moving on to the next step.

It is possible to leverage object detections at test time to boost performance without any changes to our model. Instead of conventional beam search we follow an approach similar

Method	Metrics				
	B-4	M	R	C	S
Gu <i>et al.</i> [132]	5.4	13.2	-	17.7	-
Feng <i>et al.</i> [106]	18.6	17.9	43.1	54.9	11.1
Ours	19.3	20.2	45.0	61.8	12.9
Ours (targeted)	18.7	20.9	45.1	64.1	13.5
Ours (targeted, ground truth)	19.0	21.4	45.5	70.2	14.3

Table 7.2 Comparison with the state of the art on COCO test set [188]. We evaluate under the unpaired setting of [106], using OID [221] object categories for visual concept extraction. We use beam search of size 3 and in the last two rows we further constrain it on target words (object categories) for an additional improvement.

to [10] to enforce certain words to be included in the output sentence. We refer to this experiment as targeted beam search and report results in the last two rows of Table 7.2. In particular, during beam search decoding we set object class names as target words and pick the highest probability sentence that satisfies the inclusion criteria. We get optimal results when constraining the generation with two target categories—those that were predicted with the highest probability. We further see a significant improvement when assuming a “perfect” object detector, *i.e.* using *ground truth* categories (MSCOCO annotations) instead; in this case we pick the two categories that cover the maximum area in the image.

Qualitative Results

We show qualitative results of our approach in Figure 7.6 (alignment-only model) and Figure 7.7 (joint model).

Alignment Model In Figure 7.6 we compare predicted captions between models trained with \mathcal{L}_2 -based alignment and our proposed objectives. It is important to note that in this case we only train the translation from the image domain to embedding space Φ , while the decoder is not updated from its pretrained state. We obtain higher fidelity captions, *i.e.* there is a clear improvement in the output *semantics* when training with the robust and adversarial objectives. However, the inability of the decoder to adapt to the domain shift leads to sub-optimal generation with prominent wording issues. Overall, the features seem to encode the right concepts but the decoder cannot properly describe the scene.

Joint Model Joint training of both the decoder and the translation model solves the aforementioned problem. In Figure 7.7, we show qualitative results of our joint model under the unpaired setting, trained with two different visual concept extractors: one on COCO, that uses 80 object categories and results in a total of 150k candidate captions, and one on OID, that uses 600 object categories resulting in 294k candidate captions.

Both variants describe the image contents accurately, but the OID model further benefits from the richer pool of object categories that is used for learning the alignment. An interesting example can be seen in the last image. The COCO model generates a description about a *man*. This is expected because the closest annotated object category in MSCOCO is *person*

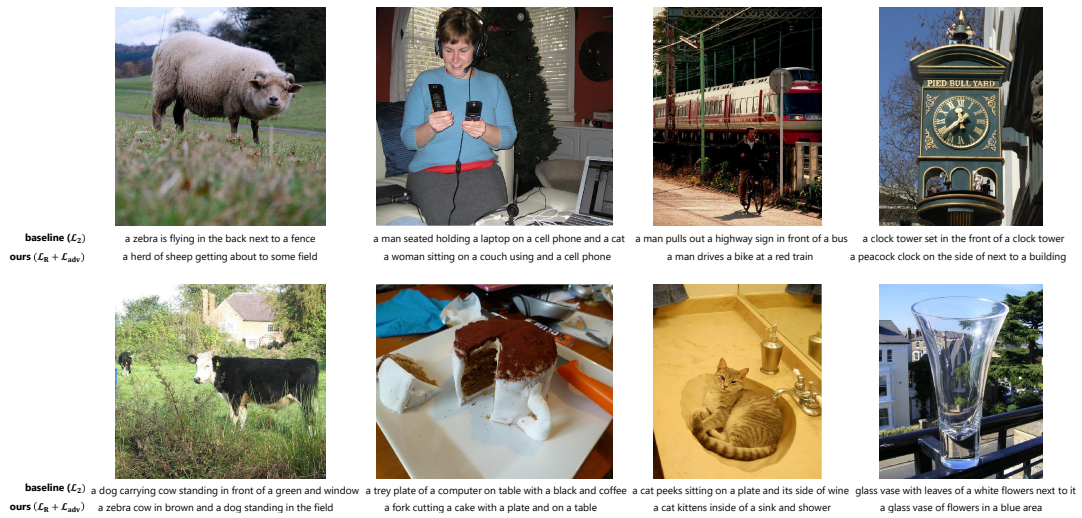


Figure 7.6 Image captioning with the alignment-only model. We show predicted captions for MSCOCO test images. For this model we only train the translation from image features to the shared embedding space but do not finetune the decoder. The proposed objectives (\mathcal{L}_R and \mathcal{L}_{adv}) perform better at this mapping than the \mathcal{L}_2 baseline.

and all its hyponyms that exist in the sentence corpus are considered equivalent. Therefore, we cannot make the distinction between *man* and *woman*, making this model prone to contextual gender bias. Since genders are distinct categories in OID, the corresponding detector allows for more fine-grained matching between images and sentences when learning the alignment. Thus, the OID model is able to resolve this ambiguity and predicts “*a woman holding a tennis racket on a tennis court*” — without using object detections at test time.

NOTE: Gender bias

Gender bias is common in MSCOCO and machine learning models, including captioning models, are known to exploit bias in the training data. This has been thoroughly studied in [18]. For example, they discuss that women are much more likely to appear in a “kitchen” than men (ratio: 0.946 for a 1:3 woman:man ratio overall in the dataset). Unsupervised models are no exception to this and in fact they rely on correlations (good or bad) which are present in the training data. As an aside — and in all cases — inferring gender from visual content alone is also problematic.

Interestingly, although the amount of known visual concepts is important, the model has the ability to extrapolate from what is labeled in the image domain thanks to common co-occurrences between different concepts. For example, in Figure 7.7 the generated words *airport*, *tracks*, *tower*, *passenger*, *grass* are unlabeled concepts that the model is able to infer from contextual information and labeled concepts such as *train*, *airplane*, *clock*, etc. This is a positive example of the model exploiting correlations present in the data, yet we do need mechanisms to overcome harmful biases that potentially come with it.



Figure 7.7 Image captioning with the joint model. We show predicted captions for MSCOCO test images. COCO and OID are results from our unpaired model, while GCC and VQA refer to the unsupervised model trained on MSCOCO images using as sentence corpus the Conceptual Captions and VQA-v2 datasets respectively.

7.4.3 Unsupervised Setting

We now move on to the unsupervised setting, where the image domain and text corpus are independent, the only link between them being a set of common concepts Ω . The choice of the language domain in this case is not trivial, as it should still contain a sufficient amount of visual concepts and descriptions. Thus in the following experiments we mostly deal with corpora that are related to visual content. During the unsupervised experiments, all hyperparameters remain the same as before.

MSCOCO/GCC/Flickr30k Quantitatively, we evaluate two cross-dataset experiments: a) MSCOCO (image domain) and GCC (language domain) and b) Flickr30k (image domain) and MSCOCO (language domain). When using GCC as the language domain, we are able to find correspondences between the image domain and approximately 1 out of 3 million captions using the detected OID categories. In Table 7.3 we report the performance of our joint model with and without adversarial training. Also here, we notice that adversarial training improves performance. However, there is still a considerable gap in performance comparing to the unpaired setting. This drop in performance does not necessarily come from bad quality descriptions, but is a by-product of the differences in vocabulary, style and context among different datasets (domain shift). Thus, quantitative evaluations in this case are not fully reliable.

We also show qualitative examples of the unsupervised setting (GCC) in Figure 7.7 and Figure 7.8. Our findings suggest that the image-caption correspondences used for training in this case are much more noisy (when compared to the unpaired setting on MSCOCO) and as the model learns to discard as much irrelevant information as possible, the resulting captions are often short. Nevertheless, the model is able to learn more abstract concepts,

Method	Metrics				
	B-4	M	R	C	W
Flickr Images ↔ COCO Captions					
Ours (w/o adv)	5.9	10.9	31.1	8.2	7.0
Ours	7.9	13.0	32.8	9.9	7.5
COCO Images ↔ Conceptual Captions					
Ours (w/o adv)	5.5	11.1	30.1	20.8	6.7
Ours	6.5	12.9	35.1	22.7	7.4

Table 7.3 Evaluation under the unsupervised setting. Image and captions come from independent sources.

which we have not observed in the unpaired setting; for example, disambiguating between a plane flying in the sky or being parked on a runway.

VQA-v2 To show the potential of our method while moving beyond combinations of captioning datasets, we also experiment with VQA-v2 [19] as the language domain. We use only the questions that are provided in this dataset as our corpus and train the model to ask questions about images. While there is not much linguistic complexity involved in this task, we find the generated questions quite relevant. We believe this can be an interesting direction regarding human-machine interaction. Qualitative results are also shown in Figure 7.7 and Figure 7.8 (VQA). We have again observed bias towards the word *man*, which seems to be the subject of the question when there is a person in the image regardless of whether this person is a man, woman or child.

J. R. R. Tolkien Finally, in a much more challenging scenario, we experiment with books by author J. R. R. Tolkien⁴ as the text corpus. We create one data point from two sentences at a time to create a more narrative style. This is a complex language domain with many rare words and long sequences, so our simple language model is not very successful at reconstructing the original inputs or learning a sentence embedding space with strong visual cues. Although all components of our model could be upgraded to accommodate for the difficulty of this task, we wish to show the different nature of the descriptions that are generated in this challenging case, in which there is only a very weak connection between the two domains (as defined by the common visual concepts). Qualitative results are shown in Figure 7.8.

7.4.4 Visualization of Embedding Space

A major component of our approach is learning an embedding space structured by a visual vocabulary. Next, we visualize this embedding space using t-SNE projections [271]. The text corpus used for these visualizations is MSCOCO captions.

⁴Available at <https://github.com/jblazzy/LOTR>

a) Language Domain: GCC



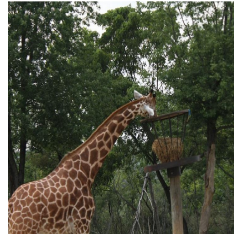
little boy playing with a teddy bear



young woman holding a glass of wine



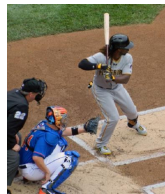
young man working on a laptop in the office



giraffe eating leaves from a tree



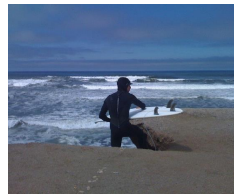
portrait of a young woman holding a tennis racket



a man in a baseball cap



sheep grazing in the snow



young man holding a surfboard on the beach

b) Language Domain: VQA-v2



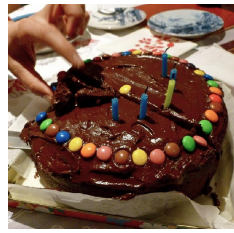
is the teddy bear wearing glasses?



are the sheep in danger?



is the man on the bench wearing headphones?



what is the shape of the cake?



is the plane landing?



what is the man holding in his hand?



is the elephant in a zoo?



is there any cheese on the pizza?

c) Language Domain: J.R.R. Tolkien



the boy drew himself up proudly. I am bergin son of beregond of the guards, he said



the houses looked large and strange to them. Sam stared up at the inn with its three <unk> and many windows, and felt his heart sink.



I know what you mean. There might be all the difference between an old cow sitting and thoughtfully chewing and an <unk>; and the change might come suddenly.



the tree rustled, and there was no sign of the elves. it was a large <unk>, and they were set to the other side

Figure 7.8 Additional results on unsupervised captioning. We show more examples from three different sentence corpora.

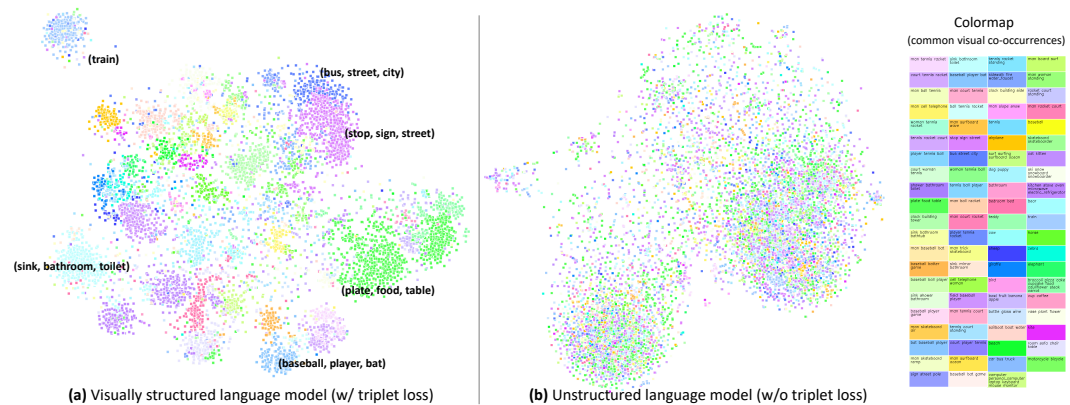


Figure 7.9 Visualization of the language model’s embedding space. We compare the t-SNE embeddings of the latent space created by our language model with a) visually structured training with the triplet loss and b) standard maximum likelihood training.

First, we analyze the *unimodal* embedding space that is created by our language model in two scenarios, as seen in Figure 7.9. We compare the standard maximum likelihood training (\mathcal{L}_{CE}) with our visually structured approach based on sentence triplets (\mathcal{L}_{CE} and \mathcal{L}_t). It is important to note that we learn the language model in an *unsupervised* manner and the categories seen in the colormap of Figure 7.9 are not used for supervision. We have manually (and not exhaustively) annotated sets of concepts that often occur together for visualization reasons only. The result verifies our intuition that standard maximum likelihood training (b) leads to an embedding space with no particular (visual) structure, while our approach (a) is clearly superior for visual tasks.

Second, we wish to check whether the *multimodal* embedding space, *i.e.* translating image features into the same space as text features, is also meaningful. In Figure 7.10 we jointly visualize the t-SNE embeddings of sentence features [L] and translated image features [I]. The latent space retains its initial visually aware structure and we further observe that the image features are well-integrated into this space, *i.e.* it is difficult to distinguish images from sentences in the feature space. In Figure 7.10 we zoom into a *baseball*-related cluster, where features from both domains are semantically close to each other. For points coming from the language domain we visualize the input to the encoder, while for points coming from the image domain we visualize a *ground truth* sentence associated with the encoded image.

7.5 Limitations and Discussion

Despite the encouraging results, image captioning without supervision is a difficult task and there are several limitations and challenges that we wish to discuss.

Maximum Likelihood Trade-off As discussed previously, joint maximum likelihood training of the alignment model and the decoder significantly boosts performance. Despite its

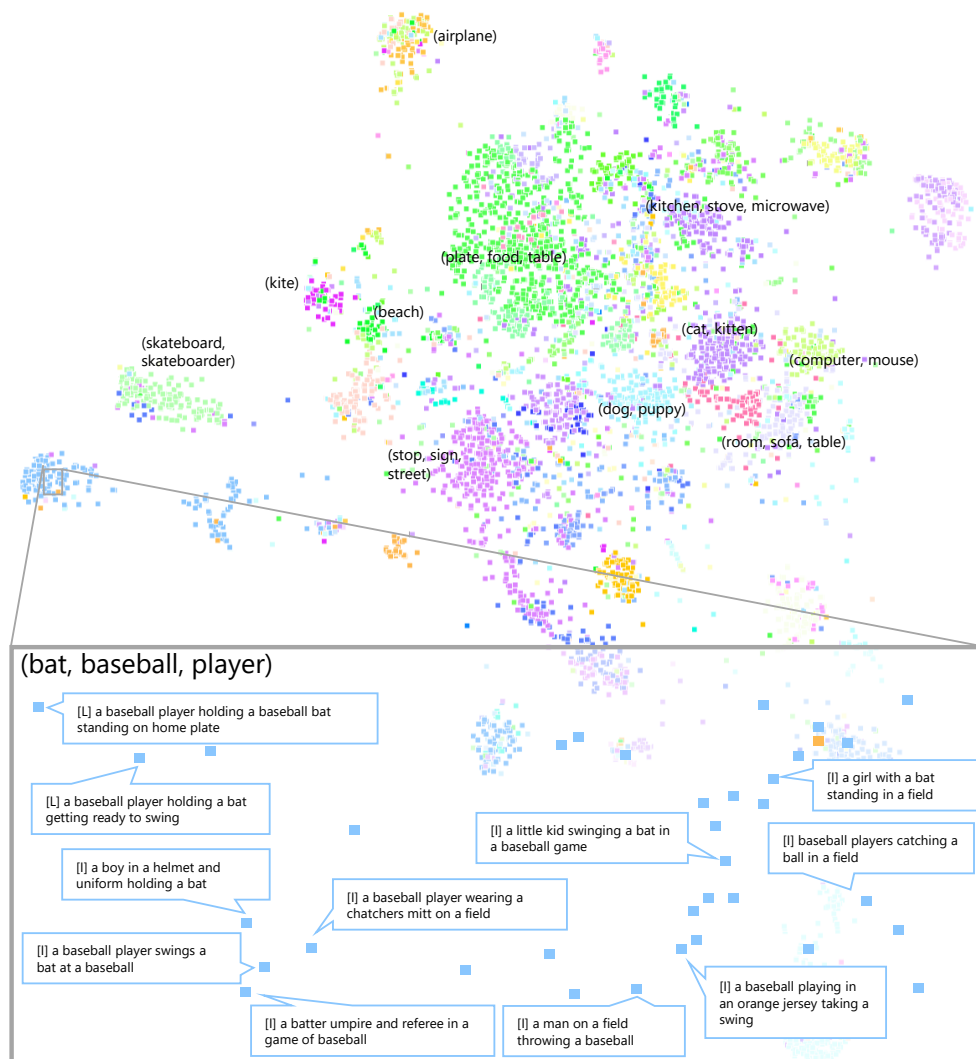


Figure 7.10 Visualization of the shared embedding space. We show the t-SNE projection of the multimodal embeddings. Zooming into one cluster we show that sentences embeddings [L] lie in visual-semantic groups with image embeddings [I].

advantage, maximum likelihood training is responsible for the rigidity of the captions in Figure 7.7. On the other hand, learning only the alignment holds more potential for naturalness and diversity but produces overall lower quality captions that suffer from both semantic and grammatical errors (Figure 7.6). Thus, we are faced with a trade-off between performance and naturalness. One solution to this problem can be reinforcement learning approaches to train the decoder in an auto-regressive mode, such as [343]. However, even those methods are usually combined with MLE.

Compositionality One major limitation is the ability of the model to describe scenes with novel or atypical compositions of objects. With respect to this, we believe there are two factors that affect the model’s behavior. Naturally, the first factor is the choice of the architecture itself, even in the supervised setting. We cannot expect unsupervised training to

overcome limitations that exist even for the supervised case. This is prominent in our case since the chosen baseline model [423] also exhibits rigid and repetitive captions and lacks in compositionality. In fact, we found that 20% of the captions generated by our model are unique and 16% are novel (not in the training set), which is similar to the findings of Vinyals et al. [423]. An object detector used during inference, similar to [265], could explicitly encourage compositionality.

The second factor is related to the *discoverable* visual concepts, *i.e.* concepts that are not necessarily labeled categories but can be anyway discovered through common co-occurrences. For instance, it is not possible to tell the difference between *a whole pizza* and *a slice of pizza*, if only *pizza* is known as a labeled concept and *slice* does not appear in any other context (in the sentence corpus). Other concepts are simply ambiguous; when a *person* and a *bicycle* are detected in the image, there are several possible scenarios about their interaction, for example predicates that link them could be *riding*, *carrying* or *parking*. This creates compositionality issues, since it is not possible to describe complex or unusual compositions without having an understanding of individual concepts or how they relate to each other. Thus, as the model learns from correlations existing in the training data, it would often produce the most probable description which is likely not a good description in the case of atypical scenes.

Intuitively, using more concepts to learn the alignment between domains allows for greater flexibility. For example, it is possible to employ relationship or attribute predictions to enrich the visual concept set on the image domain and consequently enrich the search space of sentence candidates. However, this approach would require additional supervision and thus constitutes another trade-off.

Visual Concept Overlap One last limitation of our approach is the heavy dependency on the overlapping visual concepts between the image and language domain. Nevertheless, this seems like a reasonable (and unavoidable) requirement when the goal is to describe visual scenes. To better deal with sentence corpora of different styles (*e.g.* Tolkien) that have limited combinations of visual concepts or use uncommon words to describe imagery, it is possible to exploit WordNet ontologies more extensively. Another solution to domains with limited overlap is to learn from a combination of multiple text corpora with different contents and styles.

7.6 Conclusion

In this chapter, we have focused on integrating scene understanding and natural language. In comparison to learning geometry or semantics, the intersection of natural language processing and scene understanding refers to a family of problems that are more challenging since they require holistic understanding of images, inferring abstract concepts as a composition of individual entities and being able to communicate about the visual world. The ability to communicate introduces the additional challenges from the field of natural language processing, for example related to syntax, semantics and pragmatics.

One extensively studied research problem in the intersection of computer vision and natural language is image captioning. Learning to generate image descriptions has traditionally relied on annotated pairs of images and captions. This often leads to corpora (captions) that are limited to a specific image database, factual (stating the obvious) and repetitive due to crowd-sourcing. The nature of the annotated captions introduces language priors that trained models often exploit instead of fully *seeing* and *understanding* the image.

We have presented an approach that alleviates the need for supervision, *i.e.* image-caption pairs. Our model can be trained from independent image databases and text corpora in two steps. In the first step, we train a language model on the text corpus alone and impose a “visual” structure in latent space with a triplet loss that pulls embeddings of sentences with similar visual words together and pushes dissimilar sentences apart. In the second step, our goal is to transform this unimodal embedding space into a *shared, multimodal* embedding space by embedding images into it such that they cannot be distinguished from the sentence embeddings of the previous step. If this mapping is successful, then images can be decoded into sentences by the same language model. The only supervisory signal that is used in our method is image/object labels, so that entities detected in the images can be named. These labels are used to obtain some *weak* correspondences between images and sentences to “supervise” the alignment of the two domains. We further propose a set of objectives (robust, adversarial, reconstruction) that make this possible in absence of real ground truth.

Our approach improves the state of the art on the unsupervised image captioning task, but as an unsupervised method it still falls behind fully supervised performance. Nevertheless, we believe that unsupervised (or semi-supervised) image captioning is an upcoming research direction with a lot of potential because it allows to better exploit large text corpora (of various styles) that are available on the web. Currently, the unsupervised approach has a worse understanding of what things are, which (again) leads the model to exploit language biases without really seeing. A carefully crafted semi-supervised and compositional approach could exploit the best of both worlds and potentially has the capability to surpass supervised performance.

User Interaction: Language as Input

8.1	Introduction	125
8.1.1	Motivation	126
8.1.2	Contribution	127
8.2	Related Work	128
8.3	Interaction as Feature Guiding	131
8.4	Guiding by Back-propagation	134
8.4.1	Method	134
8.4.2	Experiments	135
8.5	Guiding in Natural Language	136
8.5.1	Method	136
8.5.2	Experiments	140
8.5.3	Visualization of Guiding Vectors	145
8.6	Conclusion and Outlook	147

8.1 Introduction

In the previous chapter, we discussed the role of natural language in higher-level scene understanding, which enables the communication of knowledge between a user and a smart system. We focused on the task of image captioning, *i.e.* using natural language in the output of the system with the goal of describing imagery, which implies information flow from the system to the user. In this chapter, we will look into the reverse problem of using natural language as *input* to a visual system, thus providing spoken or written human knowledge to the system.

Indeed, there are several tasks where this is useful. Some recent examples are agents that navigate [13, 109] or execute household activities [328], based on a set of instructions that they receive in natural language which is then transformed into a set of actions to take in the environment. The approach that we present in this chapter uses natural language as input with the goal of providing *auxiliary* information to the visual system. The contents of this chapter have been published in [354].

8.1.1 Motivation

We have already seen that numerous computer vision applications are dominated by the use of convolutional neural networks, which continue to grow and push performance boundaries on various tasks. However, most approaches train models on datasets which are constructed for specific problems and it is thus likely that, when deployed in the wild, these models will face challenges due to domain shift or other unfamiliar scenarios. In practice, these models are often deployed statically inside a system and treated as black boxes. Non-expert users do not have access to or knowledge about the inner workings of the model. Therefore, when the system makes erroneous predictions, a user might observe this but still have no influence over the outcome. Now if it were possible to *dynamically* incorporate the user's feedback into the system, it could be also possible to improve its performance either in this specific instance or for the long term. We believe that for intelligent systems to be successfully deployed in the real world, this is a feature they should be equipped with.

To further motivate the idea behind our approach, imagine the example of image generation and editing. Indeed, the success of generative models (GANs) has contributed astonishing results in automatic image synthesis. Somewhat related to the topics discussed here, there exist methods for generating images from natural language descriptions [333, 339, 478]. Given the difficulty and ambiguity involved in this task, it is likely that the outcome will not be perfect and manual interaction might be needed from the user's side to refine it. The same goes for image editing. For example, in photo editing software that extracts segmentations before applying a desired operation, if the segmentation is wrong, the outcome will not be optimal. Instead of hand-drawn corrections, the user could type some hints about what went wrong to sway the outcome to their needs; alternatively they could directly request *semantic* changes by editing an intermediary scene representation known as scene graph [84].

This is the inspiration of our approach. However, there is a crucial difference to generative methods that synthesize and modify images based on iterative user requests [65, 84, 95, 372]. We instead propose interaction with systems trained for *discriminative* tasks. Starting from a model that is trained for a specific task which does not depend on user input (e.g. semantic segmentation), we introduce a module that can optionally influence the model's predictions conditioned on user input.

There exist many possibilities where this is useful, which go far beyond the image generation and editing example. For example, in high-risk scenarios such computer-aided diagnosis, the experience of medical practitioners is an asset that *must* be taken into account, especially when they disagree with the output of the system. Allowing them to interact with an algorithm in a natural way can be useful for second-order analysis tasks that, for instance, depend on a successful segmentation outcome. In another example, allowing users to interact with the algorithm can speed up cumbersome labeling tasks, because the annotators can focus on correcting major mistakes instead of providing detailed hand-drawn masks for every single object [3, 412]. A final example are *embodied agents* that are tasked with navigating in an environment, searching for objects and answering questions [73]. This is an interactive

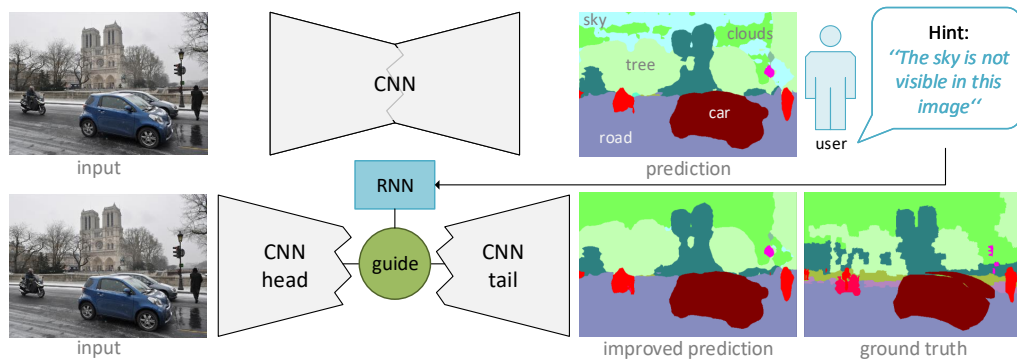


Figure 8.1 Overview of user-network interaction for semantic image segmentation. We introduce a module capable of improving the predictions of a CNN by guiding its activations conditioned on user-provided hints. The means of communication is natural language which is processed by an RNN. In this example, the base model mistakenly labels the top of the image as sky instead of clouds. When the user indicates this mistake, inference is revisited and the prediction is updated accordingly without any change of the network’s parameters.

problem in itself where the user can guide the agent to prevent it from following a wrong goal.

8.1.2 Contribution

In this chapter, we investigate the question: is it possible to still improve a model’s performance *after* it has been trained? With this goal in mind, we present a novel approach to enable interaction between a user and a deep network based on feedback given by the user at test time. While most interactive approaches position the user on the input side, *i.e.* they depend on some initialization to deliver an output, in our method user input is used as auxiliary information a posteriori. We propose a module, that we hereby refer to as the *guide*, which modulates the network’s activations spatially and semantically conditioned on optional user feedback. The means of communication is natural language (textual hints written by the user), but can be generalized to different types of input. The guide is in this case dynamic (unlike static model parameters) and acts directed by an annotator such as a human-in-the-loop.

As a testbed for the guiding idea we use the task of semantic segmentation. An overview of our method can be seen in Figure 8.1. It is noteworthy, that our method does not depend on additional annotation (besides the original segmentation task) and can thus be trained on the standard benchmark datasets. In our results we can show the proposed module is able to learn a semantic correlation between the language and the visual domain.



KEY CONTRIBUTIONS

- We propose a novel method for spatio-semantic modulation of a network's activations after the network has been trained and deployed.
- We show that, by bringing a human into the loop, the guide helps to improve performance during inference, without distracting the model from its original task, *i.e.* the model operates as intended even without the user.
- Since there exists no annotated dataset for this problem, we propose a scheme for programmatically generating user hints to train the guide.

8.2 Related Work

The approach that we discuss in this chapter is related to several fields in computer vision, namely interactive segmentation, feature modulation and dialogue-based visual tasks.

The task of semantic segmentation is extensively studied in computer vision [23, 55, 112, 252, 260, 323, 481]. Our goal is to build on established models [55, 260] and augment them with an interactive module that would allow the user (or an oracle) to request changes in their predictions.

In the following, we discuss the most prominent literature of related fields.

Interactive Image Segmentation

Human-machine interaction in the context of (semantic) image segmentation is a well researched topic. Before deep learning, popular methods for interactive image segmentation were based on graph cuts [45, 326], random walks [129], geodesic distances [25, 70], geodesic star convexity [135] and active contours [191]. The input given by the user was typically in the form of seed points [45, 326, 434], scribbles [25, 28, 129, 454], bounding boxes [130, 234, 349], yes/no interaction [54, 356], eye gaze [153], or a combination of multiple cues [458]. These are all effortless ways for a user/annotator to seed the algorithm for pixel-wise labeling. Some of these methods are based on active learning, *i.e.* the algorithm queries the user (or an oracle) [28, 54, 356, 384] about certain regions, usually based on uncertainty measures.

In the deep learning era, interaction-based segmentation has been studied mostly in the context of weakly supervised learning and has moved from foreground/background segmentation to semantic categories. The weakly labeled data most often consist of image-level labels [316, 329, 375, 430, 439], but also again user-specified clicks (seeds) [34, 210, 460], scribbles [251, 398], contours [353] or bounding boxes [72, 334]. However, this family of methods

are not active as they do not depend on a human in the loop and only require a form of annotations *once* as input to the algorithm.

Collaborative deep learning frameworks have been proposed as well; they *iteratively* alternate between human and machine and incorporate user feedback continuously to refine the output of the system while reducing annotation effort. The vast majority of these interactive frameworks deal with segmenting individual objects [1, 52, 248, 249, 272, 276, 280]. In medical imaging, where expert knowledge is required from the annotators, this is an interesting direction and several methods aim to minimize the amount of interaction required to relieve doctors from time-consuming manual labeling [8, 362, 428, 429]. Very recently, some methods have been proposed to tackle full image segmentation with interaction [3, 412], reporting performance gains over single object interactive segmentation. Similar to our approach, annotations (seeds, scribbles, etc.) are typically simulated during training, since integrating an actual user in the training loop is infeasible.

There is a major conceptual difference to our method. The aforementioned approaches are trained specifically for the task of interactive segmentation, taking an image and the user feedback as input to a CNN and predicting an output segmentation. During inference, the network parameters are fixed and the system behaves exactly as it is trained for. In our approach, we consider (a) a model which is pre-trained and fixed at test time and (b) an additional mechanism, the purpose of which is to improve the performance of the fixed model without updating its weights. As such, the guiding mechanism is not specific to semantic segmentation and can be generalized to various tasks and means of interaction. We present our method in two cases: through a learnable guiding module and guiding by back-propagation. After the publication of our approach, Jang and Kim [182] explore some similar ideas and refine predictions by back-propagating into their guidance maps. Instead, Kontogianni et al. [212] improve the network's initial predictions by interpreting user hints as training annotations and updating the model's parameters. Also related is the earlier approach of Wang et al. [429] that focuses on image-specific network finetuning at test time in the medical domain.

User interaction with machines, and in particular interactive labeling, is not limited to semantic segmentation. Some other examples include user-guided colorization of grayscale images [479], object instance localization where the system chooses the best task to assign to the human annotator [359] or fine-grained classification through interactively given attributes [39, 46].

Feature Modulation

In our method, the way we approach the problem of guiding a fixed model has its roots in feature modulation. Feature-wise transformations have been originally discussed, to the best of our knowledge in the concurrent work of Vries et al. [426] and Perez et al. [318]. The main idea is to modulate the features of a network through affine transformations given some *additional* input. This then becomes a type of conditioning or fusing information from an additional source, which has proven to be more flexible and effective than more naïve

forms of conditioning— for example, concatenation-based. This is also the inspiration for our method.

In [318, 426] feature modulation is used for VQA tasks using a language processing network (RNN) to condition the visual processing network (CNN) on the question. Perez et al. [318] propose specific layers (FiLM) to modulate CNN features and jointly train these layers with the network. On the other hand, [426] formulate this idea as conditional batch normalization [179] and learn modulation vectors while keeping the (pre-trained) network parameters fixed. Strub et al. [386] propose an improvement by iteratively switching between language processing and feature modulation, an approach that performs better for longer linguistic sequences and dialogues.

Other early demonstrations of this idea are known as *conditional instance normalization* in the context of style transfer [91, 120]. The main difference is that in this case modulation is tied to normalization layers (conditioning them on the style), which can be seen as a special case of [318]. In the same context, *adaptive instance normalization* [175] does not use learnable modulation parameters, but directly computes them from the style image using the same network architecture.

Subsequently, feature modulation has been adopted in many different tasks. Recently, generative methods have also used this type of class conditioning [47, 295]. In image recognition, it is mostly seen as self-conditioning [167, 383], *i.e.* using the network’s own activations to learn modulation vectors instead of using additional inputs.

The aforementioned methods can modulate features across channels, but lack a spatial component. Thus, in image synthesis and image-to-image translation tasks, that require high fidelity outputs, the modulating parameters vary also spatially. Bau et al. [29] learn a perturbation model to adjust the top layers of a generator to specific image content in the context of photo editing. Another example is SPADE [315], a state-of-the-art model in conditional image generation, that uses the conditioning semantic maps to guide the features of a generator, thus better preserving semantic information in the generation process. In a similar manner, Wang et al. [435] preserve texture in image super-resolution. In [463], spatio-semantic modulation is also used for video object segmentation by guiding features of the segmentation network using an annotated target object from the first frame. Feature guiding techniques are also used for few-shot learning [335] and in the context of meta-learning [301].

Dumoulin et al. [92] have contributed a detailed analysis of the mechanics and a review of the literature utilizing the feature modulation principle across different domains.

Vision-and-Language Tasks

Our goal is to allow for user interaction with deep learning models in a straightforward and natural way, which makes such systems approachable also for non-expert users. Thus, we follow an approach that uses natural language as an interface for interaction.

This makes our approach loosely related to several visual tasks that include some form of dialog. Apart from image captioning, which we have already discussed in the previous

chapter, such problems include but are not limited to: visual question answering [2, 15, 16, 169, 187, 263, 277, 278, 484], where the user poses a question about an image and the system replies in natural language; embodied question answering [73] that focuses on agents that must navigate in a real environment and understand their surroundings in order to answer the question; visual and language-based navigation [13, 109] where the agent has to follow instructions; and visual dialog, where the user engages in conversation with a bot [74, 262, 385, 425]. Finally, referring expression comprehension is the task of visually grounding phrases on images [192, 267, 473] and can be used for segmentation [171] or object retrieval [170, 425].

The main difference between our approach and most of the related literature is the final task of the system. Our output is neither an answer nor an image caption—it is visual and not textual. We do not change the task of the interactive CNN, we only augment it to allow user input. A further aspect is that we do not need to rely on vision-text correspondences such as paired questions-answers or captions. Our model simulates the user interaction via textual expressions are automatically generated.

8.3 Interaction as Feature Guiding

Next, we present two different (but related) approaches for enabling interaction between a human user and deep learning models. In Section 8.4 we exploit user-provided clicks and use them as targets for optimizing the model’s activations (but *not* its parameters) through back-propagation. In the second approach (Section 8.5), we modulate the model’s activations through learnable vectors that are conditioned on natural language inputs. By modulating activations, we are able to improve the model’s performance at a certain task *during inference* with an additional source of information—provided by the user or an oracle—without additional training samples.

Hereafter we will refer to the interaction mechanism as the *guide*. The guide receives auxiliary information about the task, which we will refer to as the *hint* and operates on the deep model’s *features*. In the general case, the hint can be an arbitrary modality with arbitrary dimensionality that might require its own processing unit (learnable or not). We insert a *guiding module* in a pre-specified layer of the network that we want to guide, which splits the network in two parts, as seen in Figure 8.1: the head $h(\cdot, \theta^{(h)})$, parametrized by $\theta^{(h)}$, which precedes the guide, and the tail $t(\cdot, \theta^{(t)})$, parametrized by $\theta^{(t)}$, that has as input the guided features for the remaining operations of the network. If x is the input and \tilde{y} is the prediction, we can then formally write this as

$$\tilde{y} = t(h(x)). \quad (8.1)$$

Guiding Module

The purpose of the guiding module is to receive hints from the user and translate them into a latent representation that is used to guide a model that is already trained for some task. We consider this model fixed, *i.e.* we do not update its weights further.

One question that arises is: what is a suitable latent representation for this purpose? The intuition behind the guide, and feature modulation in general, is that as the network encodes different task-specific aspects in each feature map of a specific layer, feature-wise re-weighting can emphasize or dampen these aspects according to an input signal.

NOTE: Feature representations

In the example of image recognition, different abstract visual concepts are encoded in the higher layers of a CNN trained for this task. Some of them can be highly correlated with one or more labeled categories. By selectively down-weighting the contribution of certain units, one could affect the ability of the network to predict the corresponding category. The field of feature visualization makes heavy use of these ideas to visualize what a network has learned, thus this re-weighting scheme is a sensible choice for the task at hand.

Let A be a feature representation inside the network defined as $h(x) = A \in \mathbb{R}^{H \times W \times C}$, where W is the width, H is the height and C is the number of channels. We define the act of guiding as an affine perturbation of A with a multiplicative vector $\gamma^{(s)} \in \mathbb{R}^C$ and an additive vector $\gamma^{(b)} \in \mathbb{R}^C$:

$$A'_c = (1 + \gamma_c^{(s)})A_c + \gamma_c^{(b)}, \quad (8.2)$$

where $c \in [1, \dots, C]$ is used to index the channels of the feature representation and the guiding representation $\gamma = (\gamma^{(s)}, \gamma^{(b)})$. γ has the role of re-weighting feature maps, or the in the extreme case acts as a switch. When $\gamma = (0, 0)$, the guiding vectors have no effect. If $\gamma_c^{(s)} = -1$ and $\gamma_c^{(b)} = 0$, feature c is suppressed with all its units set to 0. When $\gamma_c^{(s)} > 0$, the activation of feature c increases in magnitude. Values $\gamma_c^{(s)} < -1$ effectively invert feature map c by emphasizing negative elements that would have been otherwise cut-off by an activation function such as ReLU and silencing units with a positive value.

NOTE: Residual learning

Our design for the guiding module resembles the design of ResNet [151] that builds on residuals, *i.e.* we do not re-weight the feature map directly, but rather modify it with residual updates.

So far our guiding approach is similar to that of [318] and lacks a spatial component. This means that the modulation is performed uniformly across the spatial dimensions of the activation map. As a consequence, it is only possible to encourage semantic changes in the features, but not *spatial* ones. As an example a spatial change could be related to a hint that mentions the *top right* part of the image. We wish to change that to achieve more fine-grained

control that could be beneficial, especially for pixel-wise tasks, such as semantic segmentation. Thus, we extend the guiding module to include a spatial representation, similar to attention mechanisms [459]. However, inside a network, feature maps typically consist of many elements, especially in the case of architectures for high-dimensional tasks that aim to preserve spatial resolution, thus avoiding bottleneck layers. To prevent the exponential growth of guiding parameters for large feature maps, instead of element-wise control, we propose to use *vectors* $\alpha \in \mathbb{R}^H$ and $\beta \in \mathbb{R}^W$ across the vertical and horizontal dimensions of the feature map respectively. Thus, the guided activations become:

$$A'_{i,j,c} = (1 + \alpha_i + \beta_j + \gamma_c^{(s)}) A_{i,j,c} + \gamma_c^{(b)}, \quad (8.3)$$

with $i \in [1, \dots, H]$ and $j \in [1, \dots, W]$.

Then, the output of a *guided* network can be written as

$$\mathbf{y}^* = \mathbf{t} \left((1 \oplus \alpha \oplus \beta \oplus \gamma^{(s)}) \odot \mathbf{h}(\mathbf{x}) \oplus \gamma^{(b)} \right), \quad (8.4)$$

where vectors α , β , γ are first tiled and summed by \oplus and then multiplied element-wise with the feature maps with a Hadamard product \odot . We have omitted the dependency on parameters $\theta = \{\theta^{(h)}, \theta^{(t)}\}$ for simplicity.

We thus achieve a trade-off between fine-grained control and the number of guiding parameters that are required using a grid-like spatial modulation. This can be also useful when estimating the guiding vectors with a lower capacity module, where an increased number of units could lead to over-fitting.

One further observation is that in fully convolutional architectures the size of the input and consequently the size of the feature maps (H , W) can vary. Therefore we do not want α and β to depend on a fixed resolution, which can be a problem when using another network to predict these vectors — thus its output size must be fixed. Instead, since α and β represent spatial attention (and not semantic adjustment like γ), it seems reasonable to specify a preferred vector size and linearly interpolate when the the spatial dimensions of the feature map to be guided are different. This further suggests that we effectively can define any granularity level for the spatial guides.

In the following, we present two different ways of employing this guiding module in the context of semantic image segmentation. Conceptually, the guide is not task-specific and can be useful in various applications. However, depending on the task, the form of the hints and therefore the guiding module, as a processing unit, can vary.

8.4 Guiding by Back-propagation

When guiding by back-propagation we formulate the problem of finding suitable parameters α , β , γ as an energy minimization problem. While the parameters of the guide are optimizable there is no learning involved.

8.4.1 Method

Let (x_k, y_k) be a training pair, where x_k is the input and y_k is the target, and $k = [1, \dots, N]$, with N being the total number of training samples. Then, consider a deep learning model trained for a certain task by minimizing a task-specific loss $\mathcal{L}(\cdot, \cdot)$:

$$\theta^* = \operatorname{argmin}_{\theta} \frac{1}{N} \sum_{k=1}^N \mathcal{L} \left(t \left(h(x_k, \theta^{(h)}), \theta^{(t)} \right), y \right), \quad (8.5)$$

where $\theta = \{\theta^{(h)}, \theta^{(t)}\}$. In the task of semantic segmentation, \mathcal{L} can be the cross-entropy loss and the segmenter can be any state-of-the-art model, *e.g.* [55].

After the model has reached its optimal state θ^* , it is unlikely that it can deliver results with zero error, especially for unseen samples. Our goal is now to find guiding parameters that improve the network's output, *while keeping the model parameters θ fixed*. Let \hat{y} be a given hint. For example, in semantic segmentation, the hint could be user-specified click(s) that indicate the correct category for one or more pixels in the image. Then \hat{y} can be represented similarly to ground truth y , but sparse. We can then optimize the previous objective, towards parameters α , β and γ instead of θ :

$$\alpha^*, \beta^*, \gamma^* = \operatorname{argmin}_{\alpha, \beta, \gamma} (\hat{m} \odot \mathcal{L}(y^*, \hat{y})), \quad (8.6)$$

where y^* is given by Equation (8.4). Optimization here plays a corrective role instead of learning. In the above equation, \hat{m} is a binary mask that indicates the spatial coordinates where a hint is given. Masking is necessary to optimize the objective only based on this additional source of information and prevent the contribution of unknown parts to the optimization process. The guiding parameters, which lie between head and tail, are initialized to 0, which makes the outcome equivalent to the baseline segmenter.

We then only optimize the guiding variables for the *current*, single input x and hint \hat{y} , while the network's weights remain constant. Conditioned on the hint, this minimization converges to vectors α^* , β^* , γ^* , resulting in an overall improved prediction.

Since all components discussed this far are differentiable, the optimization of Equation (8.6) can be achieved via gradient descent and back-propagation.

As we later show in our experiments, the optimization process can be repeated for multiple hints, *i.e.* the user can iteratively give seed-based hints and the guiding parameters are al-

ways initialized by their previous state (initial state being $\alpha = \beta = \gamma = 0$). This allows for boosting the segmentation accuracy even further.

Guiding by back-propagation is a flexible method that can be applied to any network at any layer. It does not require any further training. However, it does require several gradient steps until convergence, which is why in Section 8.5 we adopt a learnable approach.

After the publication of our approach, a few recent methods have investigated the back-propagation technique for interactive image segmentation. Instead of optimizing modulation parameters by back-propagation, in [182], they update interaction maps that are fed as input to the baseline model. In [212] they directly update the model’s parameters to adapt to a test image/sequence based on the user hints and masked objective.

8.4.2 Experiments

Next, we evaluate the effect of guiding by back-propagation on a pre-trained CNN with fixed parameters. As a baseline model we use a fully convolutional architecture (FCN-8s) [260] trained on the PascalVOC 2012 dataset [99] for semantic segmentation. The architecture is built on top of a pre-trained VGG-16 [378] with up-sampling layers. The mean intersection over union (mIoU)¹ of this model on the PascalVOC 2012 validation set is 62.6%.

We select the layer with the lowest spatial resolution, *i.e.* bottleneck, as the guiding location. We then modulate the activations of that layer with spatial attention vectors α and β , as well as with semantic multiplicative vectors $\gamma^{(s)}$ and biases $\gamma^{(b)}$. We initialize all vectors to zeros.

To enable interaction with the network we follow a scheme similar to the 20-question game of Rupprecht et al. [356]. First, given an input image, the network performs a forward pass resulting in an initial estimate. Depending on the outcome, the user can directly click on certain points and specify their semantic category (if mistaken) — thus providing seeds for the back-propagation algorithm. Here, instead, we allow the network to first enquire the user about the class of single pixels for which it is most uncertain. We define uncertainty as the difference in the output probability between the two most confident categories for each pixel. Therefore, the pixel with the smallest difference between the top two classes is the most uncertain one, because it is the most likely one to change its label. While a real user can in theory answer the model’s enquiry, for automating the evaluation we use the ground truth provided by the dataset to specify the correct label for each queried pixel. In most cases, these pixels will correspond to object boundaries. After the answer has been provided, a backward pass is performed and only the guiding parameters are updated until convergence (usually around 200 steps). The process can be repeated multiple times, updating the guiding vectors further with each queried pixel.

In Table 8.1 we report the performance after 0, 1, 5, 10, 15 and 20 queries, with 0 denoting the base model prior to guiding. After just 20 interactions, we observe an improvement in

¹averaged over 20 classes


#questions	0	1	5	10	15	20
mIoU	62.6	65.3	73.1	76.9	77.3	81.0
pixel accuracy	91.1	91.8	94.1	95.3	96.0	96.3

Table 8.1 Performance after a number of pixel queries. We apply guiding by back-propagation on a pre-trained FCN-8s [260] on the PascalVOC 2012 val set [99]. We evaluate the model in terms of mean intersection over union (mIoU) and pixel accuracy after a number of interactions, observing constant improvement.

mIoU of nearly 20 points. Please note that the architecture we have chosen for this experiment is not the currently best performing method on the leaderboard of PascalVOC 2012², which is nearly saturated in terms of performance. One of the current top entries, DeepLabv3+ [56], reaches 87.8% in mIoU. However, our goal with this experiment is not to achieve state-of-the-art performance but to show the performance gain that is achievable through simple interactions and guiding by back-propagation. This finding can be extremely useful in human-machine collaboration for accelerating labeling tasks.

8.5 Guiding in Natural Language

Guiding by back-propagation is a straightforward way to improve a model’s predictions that does not require any further training and can thus be applied to any architecture out-of-the-box. However, the hint is expected to be in the same domain as the output and, although this is easy to achieve when it comes to semantic segmentation, it might not be generalizable to other problems or settings. Further, as previously discussed, a desirable property for real-world intelligent systems is the ability to communicate with human users. Hence, in the following, we extend the idea of guiding to a natural language interface between the deep learning model and the user. The guide then becomes a learnable module that can be plugged in on top of any pre-trained model and enables *auxiliary information in natural language* to flow from the user to model.

 **NOTE:** *Natural language: spoken or written?*

Although here we address this problem with written hints, it is possible to ease communication even further through speech and voice-to-text tools (or directly designing an audio-encoding module).

8.5.1 Method

The following sections detail the additional model that we learn to translate from textual hints to the guiding parameters of the model. Further, we will show a way to automati-

²PascalVOC 2012: segmentation challenge with additional data

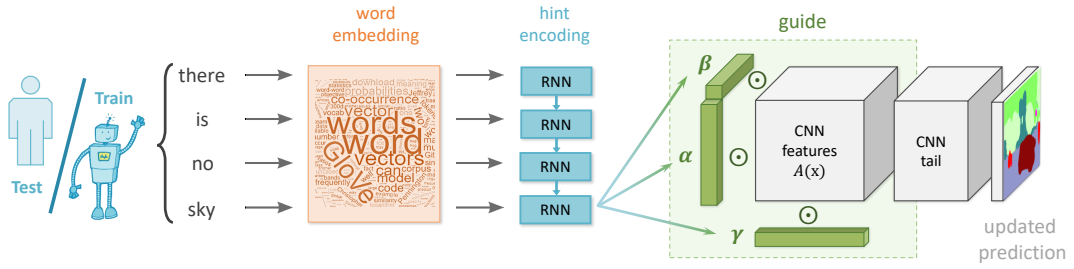


Figure 8.2 Guiding a network with natural language hints. The word embeddings of the hint sequence are encoded by an RNN, producing the guiding vectors α , β , γ (biases omitted from the visual illustration for simplicity). The guiding vectors modify the CNN features of a chosen layer, thus forcing a change in the final prediction, without any other parameter update. While a real user can interact with the network at test time, during training we simulate the user by programmatically generated hints.

cally generate synthetic hints during training such that no task-specific human annotation is necessary.

Training with Natural Language Hints

Similar to the previous chapter, we use a mapping function $f(s, \theta_f) \in \mathbb{R}^D$ to encode text sequences s into a D -dimensional embedding. $f(s)$ can be again modeled as a simple RNN. We use a word embedding matrix, pre-trained on a large corpus, to embed each word into a fixed-length representation before feeding it to the RNN. A linear layer follows the final hidden state of the RNN to map the encoded sequence into the guiding vectors α , β and γ with projection matrix $W \in \mathbb{R}^{D \times (H' + W' + 2C)}$:

$$[\alpha; \beta; \gamma^{(s)}; \gamma^{(b)}] = W^T \cdot f(s), \quad (8.7)$$

where $[\cdot; \cdot]$ denotes the concatenation of the vectors. We use H' and W' to emphasize that the dimensionality of the spatial vectors α and β does not need to match that of the feature maps (H, W) , and it suffices to linearly interpolate each vector before any element-wise operation with the features. Since the textual hints are usually short, this simple language modeling approach is enough for producing the guiding parameters.

To train the guiding module, we first make one forward pass through the base model, without the guide, which gives an initial prediction \tilde{y} . Then, a natural language hint is automatically generated with a process that we will describe shortly. The hint can be for example a phrase such as “*there is no sky*”. We denote as \hat{y} the semantic category that is associated with the hint; in the previous example, that is *sky*. The guiding parameters are predicted by processing this hint and modulate the activations of a specific layer of the base model resulting in a new *guided* prediction y^* . We train by optimizing the following objective:

$$\mathcal{L}_g(y, \tilde{y}, \hat{y}, y^*) = M(\hat{y}, \tilde{y}, y) \odot \mathcal{L}(y^*, y) \quad (8.8)$$

where M is weight mask with

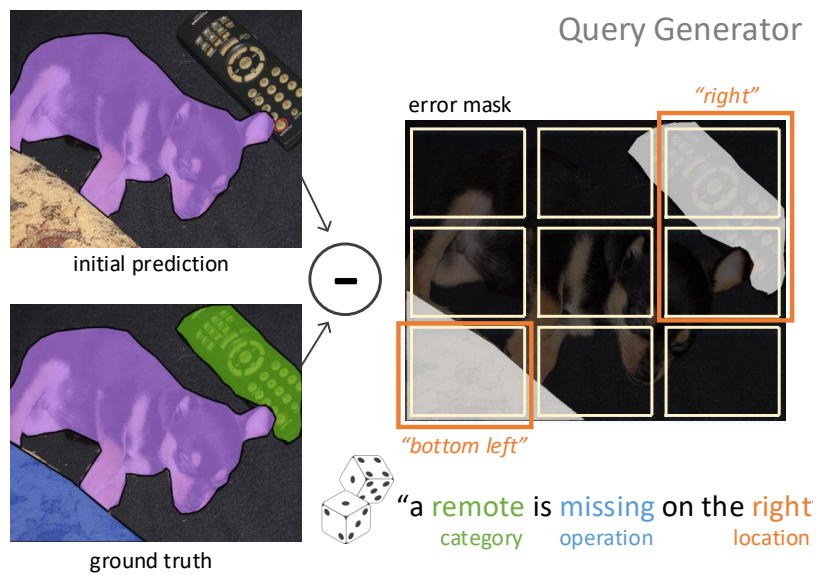


Figure 8.3 Query Generator. We illustrate the process to automatically generate queries to substitute the user during training.

$$M(\hat{y}, \tilde{y}, y) = \begin{cases} 1, & \text{if } \hat{y} = y \\ 0.5, & \text{if } \tilde{y} = y \\ 0, & \text{otherwise.} \end{cases} \quad (8.9)$$

The goal here is twofold; we want the guide to improve the performance of the base model, but also to condition changes on semantic categories. Thus, we weigh the loss with a factor of 1 for pixels whose ground truth label matches the mentioned semantic category ($\hat{y} = y$). The role of this mask is crucial in training the guide, as it forces the alignment between the verbal categories (*i.e.* what is mentioned in the hint) and the visual categories (*i.e.* output labels). Only then the guide can learn to make improvements which are actually *conditioned* on the hint. We also allow pixels whose label was correctly predicted in the first place ($\tilde{y} = y$) to contribute to the loss with a reduced factor of 0.5. This discourages the corruption of correctly labeled parts of the image. Finally, pixels that were wrongly predicted but not mentioned in the hint do not contribute to the loss, to avoid polluting the gradient signal with improvements in unrelated semantic categories.

Simulating User Hints

We will now explain how to automatically generate natural language hints; this is necessary to mimic the user during the training phase. While most work on the intersection of vision and language relies on crowd-sourced annotations—*e.g.* captions, referring expressions, questions/answers—we argue that in this case “interaction” annotations are not only difficult to acquire but also require strong assumptions about the base model’s errors. Instead, we propose a scheme for automatic generation of textual hints from vision-only

labels, *i.e.* by comparing the initial prediction and the ground truth labels at each training iteration. This allows to adapt the hints online, while training the guide, according to the behavior of the model.

To generate the hints, we create a set of sentence templates with placeholders that correspond to *functional operations*, *semantic categories* and *spatial cues*. An example can be seen in Figure 8.3.

Naturally, the functional operations are task-specific. In the case of semantic segmentation, we define and use two directions that can improve the outcome. The first one is to find and label visual categories that were initially missing from the output, but are part of the ground truth segmentation. These are the categories for which:

$$(v \in y) \wedge (v \notin \tilde{y}), v \in \mathcal{V} \quad (8.10)$$

\wedge denotes the logical and operator and \mathcal{V} is the set of visual categories. We later refer to this operation as `find`. The second type of hint is related to suppressing spurious predictions or removing wrongly predicted labels, that is

$$(v \notin y) \wedge (v \in \tilde{y}), v \in \mathcal{V} \quad (8.11)$$

We will refer to this type of hint as `remove`. Removal hints indicate the visual category that should not be there, but do not specify which class to replace it with. Nevertheless, as we also show later this is often possible to infer from image context.

We first extract all erroneous labels, ignoring those that do not meet a certain area threshold (amount of pixels they cover). We then sample a visual category from this set with probability proportional to its potential for improvement. This means that larger errors (in area) will be sampled with higher probability. To fill in the slots in the template sentence we use the semantic name of the sampled category and the associated operation, *e.g.* creating the hint “*a remote is missing*” in Figure 8.3. Then, we attempt to narrow down the error spatially. We divide the output into a 3×3 grid and depending on the grid cells that the errors falls in, we verbalize the location using one of the following phrases: “*on the top*”, “*on the top left/right*”, “*on the bottom*”, “*on the bottom left/right*”, “*on the left*”, “*on the right*”, “*in the middle*”. When the location cannot be uniquely defined, *e.g.* because the same category covers more than one region, then we omit spatial indications from the hint.

To summarize, we have now shown that it is possible to *automatically* generate sentences to provide semantic information about the image that the network has initially missed. Crucially, this is done using only existing vision labels instead of manually annotating sets of images/predictions/hints, which would be costly and inefficient. However, at test time any person (even non-experts) can take the place of the simulated user.

8.5.2 Experiments

We now adopt a more challenging setting than the PascalVOC semantic segmentation challenge [99] so that even state-of-the-art models have room for improvement, thus making a human in the loop a justifiable choice.

To evaluate our approach, we conduct experiments on the *COCO-Stuff10k dataset* [50], which is a subset of MSCOCO [253] comprised of 10,000 images from the 2014 *train* set. COCO-stuff is split into 9,000 train and 1,000 test images and contains pixel-wise labeling of 91 things (same as MSCOCO) and 91 stuff classes. The list of categories and the corresponding colormap used in this section can be seen in Table 8.5.

Implementation Details

Base Model As our base model we use DeepLab [55] with ResNet-101 [151] as the backbone. We train the model on half of the COCO-Stuff train set, which consists of 4,500 images. The input images are of size $321 \times 321 \times 3$. Due to the challenging setting and few training data, the base model achieves a mIoU score of just 30.5%.

Language Encoder and Guiding Module We save the remaining 4,500 training images to train the guide to improve the base model conditioned on user hints. The hints — generated as previously described — are embedded using a 50-dimensional GloVe word embedding matrix [317]. The word embeddings are fed as input to a GRU with 1024 hidden units. The linear layer that follows the GRU projects the 1024-dimensional final hidden state of the GRU into a vector that depends on *the number of features* in the layer where the guide is inserted. For example, in layer `res4a`, $\gamma^{(s)}$ and $\gamma^{(b)}$ are 1024-dimensional to match the number of feature maps. The spatial resolution in the same layer is 41×41 and we choose α and β to be 41-dimensional. However, it is important to note that their sizes can be chosen freely (to represent a level of granularity), as their elements can be resized with bilinear interpolation. We train all parameters related to the guide for approximately 100k steps, while keeping the weights of the base model fixed. For optimization we use SGD with a learning rate of 0.01, momentum of 0.9 and weight decay 0.0005.

Ablation Experiments

Next, we present several experiments to quantitatively study the guiding module, the importance of its position inside the base model, the functional types of hints and the effect of repeated guiding. Similar to training, we simulate the user hints for automatic evaluation. The scores reported for all experiments are the average performance over five evaluation runs to account for the randomness involved during the hint generation.

Guiding Module We experiment with different implementations of the guiding module and compare the results of the models guided by `find-hints` in Table 8.2. We compare three variants: a) the FiLM layer of Perez et al. [318], b) a dense (3D) guidance map and c) our proposed approach of semantic and spatial guiding vectors.

By design, the FiLM layer only guides semantically, *i.e.* by recalibrating feature channels. Our approach generalizes this idea to more dimensions, thus allowing the guide to learn spatial attention as well. As seen in the following table, this further improves performance by a small margin (mIoU from 33.31% to 33.56%, while mIoU before guiding is 30.5%). It is intuitive that the biggest gain comes from semantic guiding because these are the modulation parameters that relate to the object representations. α and β mostly learn to relate the location information provided by the user to spatial attention; however, spatial hints are not always applicable, thus we observe a smaller amount of improvement. Additionally, we experiment with a dense version of the modulation parameters, *e.g.* predicting a $41 \times 41 \times 1024$ map—which does not perform better. This suggests that 2D spatial attention is perhaps over-parametrized and it is harder to learn meaningful guiding maps.

We have also tried “wrapping” the modulation layer in a residual block, as in [318], but performance slightly dropped; thus in the following experiments we apply the guiding vectors directly, without a residual block.

Guiding module	mIoU	
	w/ res-block	w/o res-block
FiLM [318]	33.08	33.31
dense	33.03	33.29
ours	33.11	33.56

Table 8.2 Guiding module variations. We report the mean intersection over union (mIoU) metric for different implementations of the guiding module. In all experiments we guide layer `res4a` using `find` hints.

Guiding Location Next we experiment with the guiding location, that is which layer is better to guide. We note that after the layer’s activations have been modified, all subsequent layers will be affected although the network parameters are not updated. From our experiments, we observe that a deeper layer (close to the prediction layer) results in local changes when guided, while guiding earlier layers affects larger regions of the output. This is a product of the receptive field. In semantic segmentation (and generally classification problems), we observe that a change in the deeper layers of the network has a more immediate effect on the prediction, presumably because deeper layers contain higher-level information. A comparison over several layers can be seen in Table 8.3. The performance increases as we move the guide closer to the prediction layer.

Layer	res3a	res4a	res5a	res5c
mIoU	32.21	33.56	35.97	36.50

Table 8.3 Guiding location. We evaluate the effect of different guiding locations (layers) in terms of mIoU performance. In all experiments `find` hints are used.



NOTE: *Guiding location depends on the task*

For tasks such as depth estimation, we often require a global change in the output (e.g. the overall metric scale) from just a few sparse measurements. Taking this into account, earlier (or bottleneck) layers are more effective when guided.

Hint Functionality Finally, in Table 8.4, we study different hint functionalities. The guide is trained only with `find` hints, only with `remove` hints or their combination. Using `find` hints alone, the simulated user points out missing classes. Instead, `remove` hints only point out that an object category should not be part of the prediction. In the former case, we essentially tell the network what we are looking for; as a result, `find` hints yield over 3 points gain in performance comparing to `remove` hints. This is because `remove` is more ambiguous than `find`. Even if the network is able to remove a class from its own prediction—following the hint—it might often not know what to replace it with. This will also become clear in the qualitative examples that follow.

Hint type	Guiding location	
	res4a	res5a
<code>remove</code>	31.53	32.56
<code>find or rmv</code>	32.22	33.73
<code>find</code>	33.56	35.97

Table 8.4 Hint functionality. We compare the performance gain when training guides with different types of hints and their combination.

Qualitative Results

We present qualitative results for user-network interaction in Figure 8.4 (using `find` hints) and Figure 8.5 (using `remove` hints). In both figures, we show both the initial prediction and the refined prediction which is the result of guided feature modulation after the user provides a hint about an erroneous label. The colormap used is shown in Table 8.5.

In Figure 8.4 we observe that the guided model is able to address several shortcomings of the base model. When given `find` hints, the guided model can successfully recover from mistakes ((e) *fridge/laptop* and ambiguities ((f) *dining table/table*), recover the segmentation of only partially predicted objects ((i) *banner*) and discover smaller scale ((a) *car*, (b) *bottle*) or heavily occluded objects ((g) *chair*) that are usually missing from the initial estimate. As a by-product of modulating the activations, some segments other than the target class might be also slightly affected. However, it is clear that the guide has learned to translate the input sentences to meaningful modulation vectors, thus the changes in the prediction are strongly related to the target class. If the target class does not actually exist in the image, then the prediction is not be affected, *i.e.* the system is robust to erroneous user input.



Figure 8.4 Qualitative results using `find` hints. The guided network is able to recover from ambiguities and difficult cases, such as heavily occluded objects, and also refine partially predicted segments.

On the other hand, `remove` hints do not specify the correct category. Sometimes the correct category can be inferred from the neighborhood or image context, *e.g.* Figure 8.5 (b)-(e). In the last rows of Figure 8.5 we see some failure cases of this type of hint, as the objects which

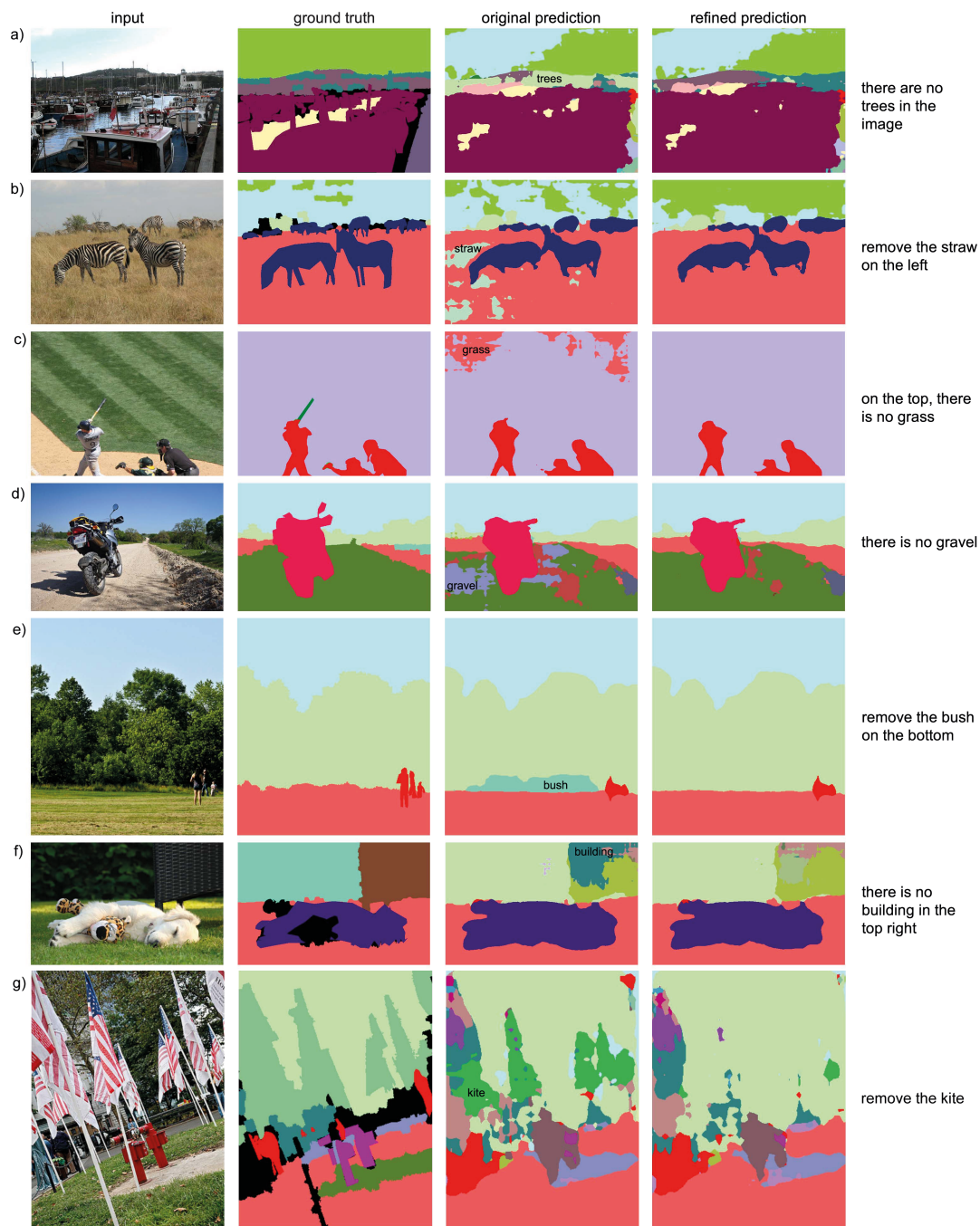


Figure 8.5 Qualitative results using remove hints. While the user specifies which category to remove, they do not specify what to replace it with. In most cases this is inferred by context.

were initially wrong can still not be recovered after the hint. The network does not have sufficient information to recognize the objects mostly due to viewpoint and thus guesses another wrong class when replacing the initial prediction.

person	bicycle	car	motorcycle	airplane	bus	train
truck	boat	traffic light	fire hydrant	street sign	stop sign	parking meter
bench	bird	cat	dog	horse	sheep	cow
elephant	bear	zebra	giraffe	hat	backpack	umbrella
shoe	eye glasses	handbag	tie	suitcase	frisbee	skis
snowboard	sports ball	kite	baseball bat	baseball glove	skateboard	surfboard
tennis racket	bottle	plate	wine glass	cup	fork	knife
spoon	bowl	banana	apple	sandwich	orange	broccoli
carrot	hot dog	pizza	donut	cake	chair	couch
potted plant	bed	mirror	dining table	window	desk	toilet
door	tv	laptop	mouse	remote	keyboard	cell phone
microwave	oven	toaster	sink	refrigerator	blender	book
clock	vase	scissors	teddy bear	hair drier	toothbrush	hair brush
banner	blanket	branch	bridge	building	bush	cabinet
cage	cardboard	carpet	ceiling	ceiling tile	cloth	clothes
clouds	counter	cupboard	curtain	desk	dirt	door
fence	marble floor	floor	stone floor	tiled floor	wooden floor	flower
fog	food	fruit	furniture	grass	gravel	ground
hill	house	leaves	light	mat	metal	mirror
moss	mountain	mud	napkin	net	paper	pavement
pillow	plant	plastic	platform	playing field	railing	railroad
river	road	rock	roof	rug	salad	sand
sea	shelf	sky	skyscraper	snow	solid	stairs
stone	straw	structural	table	tent	textile	towel
tree	vegetable	brick wall	concrete wall	wall	panel wall	stone wall
tiled wall	wooden wall	water	water drops	window blind	window	wood

Table 8.5 Semantic categories of COCO-Stuff. We visualize all semantic categories in the dataset as well as the colormap used in the remaining figures.

Can we Guide Repeatedly?

As in Section 8.4.2 (Table 8.1), we are interested in studying the effect of repeated guiding of the network with sequential hints. We guide layer `res5a` with `find` or `rmv` hints and report the performance for increasing number of hints in Table 8.6. With every hint we tackle a mistake of the most recently produced output. We find that two hints yield the best results. From there on, we see a drop in the performance gain. At this point, it is important to note that the guide is only trained with perform a single update. The modulated activations must then still be “recognizable” by the fixed parameters of the network, otherwise the prediction would not improve. However, if we guide the network multiple times certain features might get modulated beyond values that subsequent layers can “tolerate”, which explains the drop in performance gain. This can be mitigated by also using sequential hints when training the guide. A qualitative example of repeated guiding is shown in Figure 8.6.

8.5.3 Visualization of Guiding Vectors

Finally, it would be visualize the embedding space that the guiding vectors form, in particular the semantic vector γ . Intuitively, these vectors must primarily encode the object

# hints	0	1	2	3	4
mIoU	30.53	34.04	35.01	34.24	31.44

Table 8.6 Repeated guiding. We guide the network repeatedly with several `find` or `rmv` hints, each one depending on the latest prediction. After three hints, the performance gain decreases because the guide, which has been only trained with a single hint, over-modulates the features.

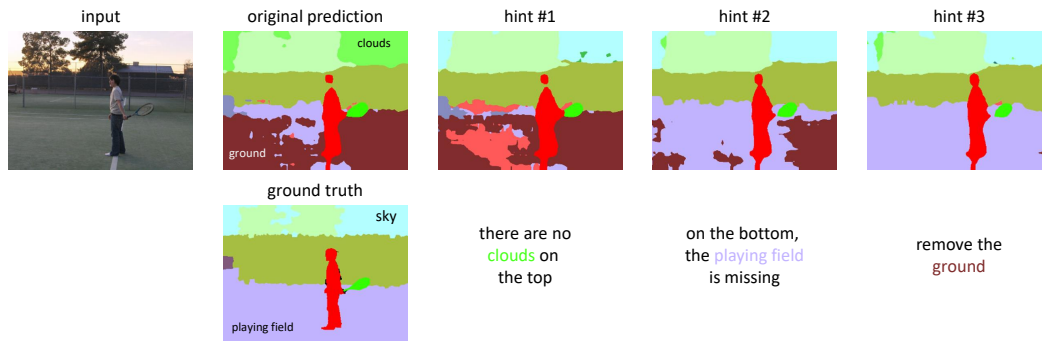


Figure 8.6 Qualitative example of repeated guiding. We provide multiple hints to the network to keep correcting mistakes with respect to confused classes sky/clouds, playing field/ground.

category that is mentioned by the user. Thus the vectors should be distinct for different object categories and likely form clusters with related object categories. In other words, if we assume that visually similar categories (*e.g.* *laptop* and *keyboard*) also have similar feature representations, then their guiding vectors must also be similar (for the same functional operation).

We visualize the t-SNE projection [271] of the γ -vectors in Figure 8.7. Each point represents the γ -vector of each object category in the dataset (182 in total) as predicted by the guide based on the same `find` hint, *e.g.* “there is ...”. The colors used to label the points in the figure represent super-categories and they are only used for visualization purposes. We easily observe that the guiding vectors of *semantically* similar words cluster together, which is an indication that the guide has learned to correlate the verbal categories with their visual-semantic counterpart. Categories that lie very close to each other have similar guiding vectors—these are categories that the network cannot easily disambiguate (*e.g.* *mud* and *sand*) either before or after guiding.

In addition to the semantic vectors, we also visualize the combination of all three vectors, including the spatial dimensions, in Figure 8.8. Here we tile and average the three guiding vectors to form a 2D map for visualization purposes. This essentially represents the magnitude of feature modulation across spatial locations. In the example shown in the figure, we guide the network with two hints. In (b), we show a hint that leads to a failure case—the guided network is still not able to find the surfboard after the user hint. Although the heatmap suggests that the correct region of the image is targeted by the guiding vectors, the change in activation might not be enough to cause a change in the output.



Figure 8.7 Visualization of γ -vectors. We show the t-SNE plot of the learned semantic vectors for all categories in COCO-Stuff. In this plot the same `find` hint is encoded for each category. The colormap corresponds to super-categories.

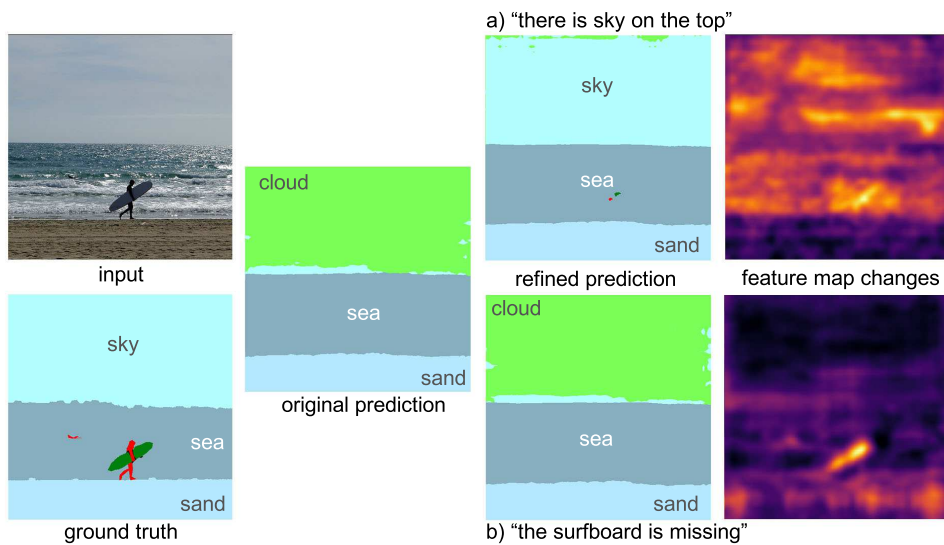


Figure 8.8 Visualization of all guiding vectors. We tile the guiding vectors to visualize them as a 2D heatmap. Two examples are shown: **(a)** Given the hint, the refined prediction is correct. **(b)** The missing category is not recovered, even after the hint, although the heatmap suggests that the guide has the right focus.

8.6 Conclusion and Outlook

In this chapter we have discussed a guiding mechanism that enables a human user to interact with deep networks. Our main goal is to enable the interaction in a most natural way, for example through human language; but as we have shown earlier in this chapter other forms of interaction are possible, for example user clicks, or scribbles.

Specifically, we address the problem of semantic segmentation. When augmenting a segmentation framework with a guiding mechanism we can think of the outcome as interactive segmentation. This can be a particularly useful approach to reduce human labor in the context of image labeling, which is necessary for training supervised learning algorithms. In this case, the machine provides an initial estimate to the annotator. The annotator's role is to help the machine improve upon its initial prediction and, therefore, it is not required to hand-draw detailed object contours from scratch.

The goal of our work is to pave the way towards interactive agents, though the current system does not come without limitations. First, the simulated user hints presented so far are restricted in terms of variability. In practice, real users might use different phrases for the same goal or even refer to a concept with different words. If the guide has not been exposed to these (*e.g.* as opposed to a language model trained on rich text corpora), it will most likely not be able to understand new vocabulary or sentence structure. Thus, in the future, apart from simulated hints, text corpora must be considered to learn a proper language model in addition to the guiding task. Another way to enrich hint vocabulary and variability is by using *instance* annotations. We can then distinguish between different instances of the same class, some of which might be erroneous while others are accurately predicted. This would allow to generate more specific hints, such as “*there is a person lying on the bed*” — to selectively refer to this particular person, when there are more people in the room.

Another limitation is that for an agent deployed in a real environment, interacting with the user can be a way to experience new concepts. Thus, we ideally want the agent to *learn* from this experience over time, instead of improving its performance only momentarily during the interaction. This is also the goal in the field of *active learning* [370], where the agent queries the user (or an oracle) to obtain information while exploring its environment. More recently, some methods have proposed to learn by asking questions [294, 462].

Overall, we deem this as an important future direction with practical impact in various fields. For example, in medical imaging it can give physicians a means to influence a computer system and directly incorporate expert knowledge — prior to subsequent analysis of the data. Interaction with the user, as well as learning from it, is also a powerful tool for embodied agents [73] and merits future research.

Part IV

Conclusion

Conclusion

9.1 Summary of Findings	151
9.2 Future Outlook	153

9.1 Summary of Findings

In this dissertation we have addressed various topics related to scene understanding which are all considered as the foundation for supporting higher-level intelligence. We will now revisit the three principles that we introduced in Chapter 1 to group and summarize our contributions.

Perception

For a system to be able to act intelligently, it should be first of all capable of perceiving its environment. This means that it is vital not only to receive some sensory input (*e.g.* image data), but more significantly to be able to interpret this information and develop an understanding for what is taking place in the surrounding environment—hence *scene understanding*. For example, this is a necessary first step for an agent to navigate in the environment and interact with objects that lie in it.

We address some of the core goals of scene understanding in Chapters 4 and 5. We study depth estimation from single images as an alternative (or complementary) to depth sensing. This is an important field of research which has attracted a lot of attention in computer vision, together with depth completion. It allows to predict dense depth maps of the environment where sensors might yield noisy or sparse measurements or just be too costly or bulky to use. We address the problem of depth estimation with a deep architecture, which at the time of publication [227] delivered state-of-the-art performance. Since then, the proposed architecture and objective loss function have been adopted by future work that builds upon our initial publication. While this method addresses scene understanding from a geometric standpoint, it provides only partial information about the environment. The second part comes, as is also the case for the human brain, from semantic knowledge; thus we also extend the method to address the task of semantic scene segmentation.

Then, to build a basis for problems such as navigation and localization, we integrate the learned geometric and semantic cues into a hybrid SLAM framework. Crucially, we build a unified framework in which we leverage the complementary nature of the learned models and traditional monocular localization and mapping techniques to better tackle the challenges that arise in each approach when addressed individually. The outcome is a dense,

real-scale 3D map of the environment that additionally holds meaningful semantic information. This model represents the basis for a holistic understanding of the scene as it includes the 3D structure, camera pose and objects contained within.

Last but not least, we conclude the first part of the dissertation by moving from a scene-centric down to an object-centric viewpoint and address scene understanding applications in the context of robotic tools or arms (Chapter 6). We show that additional information can be recovered when analyzing objects for their physical affordances, for example predicting how to grasp individual objects present in the scene.

Communication

In the second part of the dissertation, we emphasize the ability of an agent to communicate in human language bidirectionally, *i.e.* to both speak and understand it. We deem this essential for two reasons. First, it might be required by the task itself that the agent is designed for; this is for example the case in conversational agents, home assistants, systems to aid visually impaired users or assistive technologies in general. Second, it is easier for such systems to be adopted by society and grow if they can communicate their understanding or their decision making process, hold conversations or ask questions about their environment, making them more user-friendly.

It is clear that the last example relies on the ability to perceive as well. Fortunately, as rapid progress has been made over the last years in the field of computer vision, visual recognition has now matured enough to open the door to such inter-disciplinary problems; *e.g.* combining disciplines in vision and natural language processing. Thus, in the second part of the thesis, we are interested in problems that lie on the intersection of these fields. One such problem is image captioning, which is a generative task that requires joint understanding on both image and language domains, *i.e.* recognizing objects and their interactions while at the same time generating coherent sentences. Due to the difficulty of the task, most progress has been made on curated datasets of manually labeled image-caption pairs. In Chapter 7, we present an alternative approach and address image captioning in an unsupervised manner; this allows to exploit large unpaired sources of images and text corpora. The only form of supervision comes in that we use visual recognition tools (*e.g.* an object detector) to identify and name the entities that are present in the image domain, bridging the gap to semantics (*i.e.* the *meaning* of things). Our findings suggest that it is possible to map images to captions even without training pairs thanks to common concept co-occurrences in both domains. However, we believe that this problem would significantly benefit from combining available unpaired sources with some additional constraints coming from paired data.

Interaction

Finally, an agent that is made to act in human environments will most certainly require the ability to interact with humans, which is of course also dependent on the previous point, *i.e.* the ability to communicate. As an example, one can think of an agent deployed in the real world; it will inevitably come across objects or situations it does not recognize, either because of unfamiliar environmental conditions (*e.g.* it has never seen this object before in

a similar scene or from this viewpoint) or because it encounters a new concept altogether. In these cases, the agent can ask the human for help — and potentially even learn from this experience.

In Chapter 8 we discuss the potential of a mechanism that enables a human user to provide feedback to a deep model during inference. Crucially, the model initially does not rely on input from the user to perform the task it is designed for. The interaction mechanism is introduced to guide the model’s behavior retrospectively, especially when it makes a mistake, and the goal is to improve the output. With respect to the form of interaction, we see various possibilities, some of which are task-dependent. Perhaps one of the most natural ways, though challenging for a computer, is interacting in human language; this is the approach we follow here. We use the task of semantic segmentation as testbed, but the idea of guiding a model is general enough and applicable to other problems as well — for example, revisiting the problem of depth estimation, an interesting application is to guide the recovery of scene geometry with sparse depth inputs or semantic knowledge.

9.2 Future Outlook

We live in an era that sees tremendous, fast-paced achievements in AI which transform technologies in various fields and several aspects of society; from unprecedented progress in machine translation to self-driving cars to solving global environmental challenges. As a personal viewpoint, the future of AI is grounded in research that lies on the intersection of computer vision, machine learning and natural language processing.

In this dissertation, we have addressed problems which deal with the ability of machines to perceive their environment holistically and communicate their understanding to human users. Developing such systems has not only theoretical but also practical implications, as they are being deployed into the real world, in autonomous driving, medicine and assistive technology. One future direction would be to study how different problem definitions can influence each other and address all related tasks in a unified framework that allows us to exploit knowledge from multiple domains.

However, currently it is often the case that methods are developed under “sterile” conditions and curated benchmarks. As significant progress has been now achieved in supervised settings, it is important to start addressing problems *in the wild* and draw attention towards hybrid approaches — supervised, unsupervised and reinforcement learning. Unsupervised learning is an important future direction because it enables learning strong representations from large, readily available data sources without the need to explicitly annotate everything in the world.

Defining and annotating all possible situations that an agent might encounter is indeed infeasible. As intelligent agents move into the real world, they will most likely come across visual objects, semantic or physical properties and relationships that they might not recognize. For this reason, the visual processing machinery of such agents must be equipped

with mechanisms that enable lifelong learning and common sense reasoning. In Chapter 8 we discussed an interaction mechanism between human users and machines. As part of future research we would like to extend this mechanism to active learning so that the agent will pose enquiries to the user, when in uncertain situations, and continue to grow through this interaction.

I hope that the methods presented and discussed in this dissertation will evolve and inspire future research towards holistic scene understanding and the development of increasingly intelligent machines.

Authored and Co-authored Publications

Authored

I. **Laina**^{*}, C. Rupprecht^{*}, V. Belagiannis, F. Tombari, and N. Navab. “Deeper depth prediction with fully convolutional residual networks”. In: *2016 Fourth international conference on 3D vision (3DV)*. IEEE. 2016, pp. 239–248.

C. Rupprecht^{*}, I. **Laina**^{*}, N. Navab, G. D. Hager, and F. Tombari. “Guide me: Interacting with deep networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8551–8561.

I. **Laina**^{*}, N. Rieke^{*}, C. Rupprecht, J. P. Vizcaíno, A. Eslami, F. Tombari, and N. Navab. “Concurrent segmentation and localization for tracking of surgical instruments”. In: *International conference on medical image computing and computer-assisted intervention*. Springer. 2017, pp. 664–672.

I. **Laina**, C. Rupprecht, and N. Navab. “Towards Unsupervised Image Captioning with Shared Multimodal Embeddings”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 7414–7424.

Co-authored

K. Tateno, F. Tombari, I. **Laina**, and N. Navab. “CNN-SLAM: Real-time dense monocular slam with learned depth prediction”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6243–6252

C. Rupprecht, I. **Laina**, R. DiPietro, M. Baust, F. Tombari, N. Navab, and G. D. Hager. “Learning in an uncertain world: Representing ambiguity through multiple hypotheses”. In: *International Conference on Computer Vision (ICCV)*. 2017

G. Ghazaei, I. **Laina**, C. Rupprecht, F. Tombari, N. Navab, and K. Nazarpour. “Dealing with ambiguity in robotic grasping via multiple predictions”. In: *Asian Conference on Computer Vision*. Springer. 2018, pp. 38–55

H. Dharmo, K. Tateno, I. **Laina**, N. Navab, and F. Tombari. “Peeking behind objects: Layered depth prediction from a single image”. In: *Pattern Recognition Letters* 125 (2019), pp. 333–340

H. Dharmo, A. Farshad, I. **Laina**, N. Navab, G. D. Hager, F. Tombari, and C. Rupprecht. “Semantic Image Manipulation Using Scene Graphs”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 5213–5222

Bibliography

- [1] D. Acuna, H. Ling, A. Kar, and S. Fidler. “Efficient interactive annotation of segmentation datasets with polygon-rnn++”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 859–868 (see p. 129).
- [2] A. Agrawal, J. Lu, S. Antol, M. Mitchell, C. L. Zitnick, D. Parikh, and D. Batra. “Vqa: Visual question answering”. In: *International Journal of Computer Vision* 123.1 (2017), pp. 4–31 (see p. 131).
- [3] E. Agustsson, J. R. Uijlings, and V. Ferrari. “Interactive Full Image Segmentation by Considering All Regions Jointly”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 11622–11631 (see pp. 126, 129).
- [4] F. Aleotti, F. Tosi, M. Poggi, and S. Mattocchia. “Generative adversarial networks for unsupervised monocular depth prediction”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 0–0 (see p. 30).
- [5] M. Allan, A. Shvets, T. Kurmann, Z. Zhang, R. Duggal, Y.-H. Su, N. Rieke, I. Laina, N. Kalavakonda, S. Bodenstedt, et al. “2017 robotic instrument segmentation challenge”. In: *arXiv preprint arXiv:1902.06426* (2019) (see p. 81).
- [6] M. Allan, S. Ourselin, S. Thompson, D. J. Hawkes, J. Kelly, and D. Stoyanov. “Toward detection and localization of instruments in minimally invasive surgery”. In: *IEEE Transactions on Biomedical Engineering* 60, pp. 1050 – 1058 (2013) (see p. 73).
- [7] M. Allan, S. Ourselin, S. Thompson, D. J. Hawkes, J. Kelly, and D. Stoyanov. “Toward detection and localization of instruments in minimally invasive surgery”. In: *IEEE Transactions on Biomedical Engineering* 60.4 (2012), pp. 1050–1058 (see p. 73).
- [8] M. Amrehn, S. Gaube, M. Unberath, F. Schebesch, T. Horz, M. Strumia, S. Steidl, M. Kowarschik, and A. Maier. “UI-Net: Interactive Artificial Neural Networks for Iterative Image Segmentation Based on a User Model”. In: *arXiv preprint arXiv:1709.03450* (2017) (see p. 129).
- [9] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. “Bottom-up and top-down attention for image captioning and visual question answering”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 6077–6086 (see p. 99).
- [10] P. Anderson, B. Fernando, M. Johnson, and S. Gould. “Guided Open Vocabulary Image Captioning with Constrained Beam Search”. In: *EMNLP*. 2017 (see pp. 100, 116).
- [11] P. Anderson, S. Gould, and M. Johnson. “Partially-Supervised Image Captioning”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 1879–1890 (see p. 100).
- [12] P. Anderson, B. Fernando, M. Johnson, and S. Gould. “Spice: Semantic propositional image caption evaluation”. In: *European Conference on Computer Vision*. Springer. 2016, pp. 382–398 (see pp. 99, 113).
- [13] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. van den Hengel. “Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3674–3683 (see pp. 125, 131).

- [14] L. Andraghetti, P. Myriokefalitakis, P. L. Dovesi, B. Luque, M. Poggi, A. Pieropan, and S. Mattocchia. “Enhancing self-supervised monocular depth estimation with traditional visual odometry”. In: *2019 International Conference on 3D Vision (3DV)*. IEEE. 2019, pp. 424–433 (see p. 31).
- [15] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. “Learning to compose neural networks for question answering”. In: *arXiv preprint arXiv:1601.01705* (2016) (see p. 131).
- [16] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. “Neural module networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 39–48 (see p. 131).
- [17] L. Anne Hendricks, S. Venugopalan, M. Rohrbach, R. Mooney, K. Saenko, and T. Darrell. “Deep compositional captioning: Describing novel object categories without paired training data”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1–10 (see p. 99).
- [18] L. Anne Hendricks, K. Burns, K. Saenko, T. Darrell, and A. Rohrbach. “Women also snowboard: Overcoming bias in captioning models”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 771–787 (see p. 117).
- [19] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. “VQA: Visual Question Answering”. In: *International Conference on Computer Vision (ICCV)*. 2015 (see pp. 111, 119).
- [20] U. Asif, M. Bennamoun, and F. A. Sohel. “RGB-D object recognition and grasp detection using hierarchical cascaded forests”. In: *IEEE Transactions on Robotics* 33.3 (2017), pp. 547–564 (see pp. 81, 82, 86, 88).
- [21] A. Atapour-Abarghouei and T. P. Breckon. “Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2800–2810 (see p. 29).
- [22] J. L. Ba, J. R. Kiros, and G. E. Hinton. “Layer normalization”. In: *arXiv preprint arXiv:1607.06450* (2016) (see p. 19).
- [23] V. Badrinarayanan, A. Kendall, and R. Cipolla. “Segnet: A deep convolutional encoder-decoder architecture for image segmentation”. In: *IEEE transactions on pattern analysis and machine intelligence* 39.12 (2017), pp. 2481–2495 (see pp. 52, 128).
- [24] D. Bahdanau, K. Cho, and Y. Bengio. “Neural machine translation by jointly learning to align and translate”. In: *arXiv preprint arXiv:1409.0473* (2014) (see p. 100).
- [25] X. Bai and G. Sapiro. “A geodesic framework for fast interactive image and video segmentation and matting”. In: *2007 IEEE 11th International Conference on Computer Vision*. IEEE. 2007, pp. 1–8 (see p. 128).
- [26] S. T. Barnard and W. B. Thompson. “Disparity analysis of images”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 4 (1980), pp. 333–340 (see p. 26).
- [27] J. T. Barron, A. Adams, Y. Shih, and C. Hernández. “Fast Bilateral-Space Stereo for Synthetic Defocus”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 4466–4474 (see p. 49).
- [28] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. “icoseg: Interactive co-segmentation with intelligent scribble guidance”. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE. 2010, pp. 3169–3176 (see p. 128).
- [29] D. Bau, H. Strobel, W. Peebles, J. Wulff, B. Zhou, J.-Y. Zhu, and A. Torralba. “Semantic photo manipulation with a generative image prior”. In: *ACM Transactions on Graphics (TOG)* 38.4 (2019), p. 59 (see p. 130).

- [30] A. Bauer, D. Wollherr, and M. Buss. "Human-robot collaboration: a survey". In: *International Journal of Humanoid Robotics* 5.01 (2008), pp. 47–66 (see p. 6).
- [31] Z. Bauer, A. Dominguez, E. Cruz, F. Gomez-Donoso, S. Orts-Escolano, and M. Cazorla. "Enhancing perception for the visually impaired with deep learning techniques and low-cost wearable sensors". In: *Pattern recognition letters* 137 (2020), pp. 27–36 (see p. 31).
- [32] J. Baxter. "Learning Internal Representations". In: *Proceedings of the Eighth Annual Conference on Computational Learning Theory*. COLT 95. Santa Cruz, California, USA: Association for Computing Machinery, 1995, pp. 311320 (see p. 76).
- [33] H. Bay, T. Tuytelaars, and L. Van Gool. "Surf: Speeded up robust features". In: *European conference on computer vision*. Springer. 2006, pp. 404–417 (see p. 58).
- [34] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. "Whats the point: Semantic segmentation with point supervision". In: *European conference on computer vision*. Springer. 2016, pp. 549–565 (see p. 128).
- [35] V. Belagiannis and A. Zisserman. "Recurrent human pose estimation". In: *International Conference on Automatic Face & Gesture Recognition (FG 2017)*. 2017 (see p. 75).
- [36] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer. "Scheduled sampling for sequence prediction with recurrent neural networks". In: *Advances in Neural Information Processing Systems*. 2015, pp. 1171–1179 (see p. 98).
- [37] J. Bian, Z. Li, N. Wang, H. Zhan, C. Shen, M.-M. Cheng, and I. Reid. "Unsupervised scale-consistent depth and ego-motion learning from monocular video". In: *Advances in Neural Information Processing Systems*. 2019, pp. 35–45 (see p. 30).
- [38] A. Bicchi and V. Kumar. "Robotic grasping and contact: A review". In: *Proceedings of 2000 IEEE International Conference on Robotics and Automation (ICRA)*. Vol. 1. IEEE. 2000, pp. 348–353 (see p. 81).
- [39] A. Biswas and D. Parikh. "Simultaneous active learning of classifiers & attributes via relative feedback". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 644–651 (see p. 129).
- [40] D. M. Blei, A. Y. Ng, and M. I. Jordan. "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022 (see p. 102).
- [41] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison. "CodeSLAM Learning a compact, optimisable representation for dense visual SLAM". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2560–2568 (see p. 60).
- [42] M. A. Boden. *Mind as machine: A history of cognitive science*. Oxford University Press, 2008 (see p. 4).
- [43] D. Bouget, R. Benenson, M. Omran, L. Riffaud, B. Schiele, and P. Jannin. "Detecting surgical tools by modelling local appearance and global shape". In: *Trans. on Medical Imaging* 34.12 (2015) (see p. 73).
- [44] D. Bouget, M. Allan, D. Stoyanov, and P. Jannin. "Vision-based and marker-less surgical tool detection and tracking: a review of the literature". In: *Medical Image Analysis* 35 (2017) (see p. 73).
- [45] Y. Y. Boykov and M.-P. Jolly. "Interactive graph cuts for optimal boundary & region segmentation of objects in ND images". In: *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*. Vol. 1. IEEE. 2001, pp. 105–112 (see p. 128).
- [46] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie. "Visual recognition with humans in the loop". In: *European Conference on Computer Vision*. Springer. 2010, pp. 438–451 (see p. 129).

- [47] A. Brock, J. Donahue, and K. Simonyan. "Large Scale GAN Training for High Fidelity Natural Image Synthesis". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. 2019 (see p. 130).
- [48] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. "Language models are few-shot learners". In: *arXiv preprint arXiv:2005.14165* (2020) (see p. 5).
- [49] A. Bulat and G. Tzimiropoulos. "Human pose estimation via convolutional part heatmap regression". In: *European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 717–732 (see p. 75).
- [50] H. Caesar, J. Uijlings, and V. Ferrari. "Coco-stuff: Thing and stuff classes in context". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1209–1218 (see p. 140).
- [51] Y. Cao, Z. Wu, and C. Shen. "Estimating depth from monocular images as classification using deep fully convolutional residual networks". In: *IEEE Transactions on Circuits and Systems for Video Technology* 28.11 (2017), pp. 3174–3182 (see p. 29).
- [52] L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler. "Annotating object instances with a polygon-rnn". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 5230–5238 (see p. 129).
- [53] A. Cauchy et al. "Méthode générale pour la résolution des systemes déquations simultanées". In: *Comp. Rend. Sci. Paris* 25.1847 (1847), pp. 536–538 (see p. 16).
- [54] D.-J. Chen, H.-T. Chen, and L.-W. Chang. "Interactive Segmentation from 1-Bit Feedback". In: *Asian Conference on Computer Vision*. Springer. 2016, pp. 261–274 (see p. 128).
- [55] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs". In: *IEEE transactions on pattern analysis and machine intelligence* 40.4 (2017), pp. 834–848 (see pp. 52, 128, 134, 140).
- [56] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. "Encoder-decoder with atrous separable convolution for semantic image segmentation". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 801–818 (see p. 136).
- [57] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua. "Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 5659–5667 (see p. 99).
- [58] R. Chen, F. Mahmood, A. Yuille, and N. J. Durr. "Rethinking monocular depth estimation with adversarial training". In: *arXiv preprint arXiv:1808.07528* (2018) (see p. 29).
- [59] T.-H. Chen, Y.-H. Liao, C.-Y. Chuang, W.-T. Hsu, J. Fu, and M. Sun. "Show, adapt and tell: Adversarial training of cross-domain image captioner". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 521–530 (see p. 100).
- [60] W. Chen, S. Qian, and J. Deng. "Learning single-image depth from videos using quality assessment networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5604–5613 (see p. 29).
- [61] W. Chen, Z. Fu, D. Yang, and J. Deng. "Single-image depth perception in the wild". In: *Advances in neural information processing systems*. 2016, pp. 730–738 (see p. 29).
- [62] X. Chen and L. C. Zitnick. "Mind's eye: A recurrent visual representation for image caption generation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 2422–2431 (see p. 101).

- [63] Y. Chen, C. Schmid, and C. Sminchisescu. "Self-supervised Learning with Geometric Constraints in Monocular Video: Connecting Flow, Depth, and Camera". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 7063–7072 (see p. 30).
- [64] X. Cheng, P. Wang, and R. Yang. "Depth estimation via affinity learned with convolutional spatial propagation network". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 103–119 (see p. 29).
- [65] Y. Cheng, Z. Gan, Y. Li, J. Liu, and J. Gao. "Sequential attention GAN for interactive image editing". In: *Proceedings of the 28th ACM International Conference on Multimedia*. 2020, pp. 4383–4391 (see p. 126).
- [66] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078* (2014) (see pp. 20, 104, 112, 113).
- [67] F.-J. Chu, R. Xu, and P. A. Vela. "Real-world multiobject, multigrasp detection". In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 3355–3362 (see pp. 81, 84, 89).
- [68] R. Clark, S. Wang, A. Markham, N. Trigoni, and H. Wen. "Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6856–6864 (see p. 59).
- [69] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. "Fast and accurate deep network learning by exponential linear units (elus)". In: *arXiv preprint arXiv:1511.07289* (2015) (see p. 15).
- [70] A. Criminisi, T. Sharp, and A. Blake. "Geos: Geodesic image segmentation". In: *European Conference on Computer Vision*. Springer. 2008, pp. 99–112 (see p. 128).
- [71] B. Dai, D. Lin, R. Urtasun, and S. Fidler. "Towards Diverse and Natural Image Descriptions via a Conditional GAN". In: *arXiv preprint arXiv:1703.06029* (2017) (see pp. 99, 110).
- [72] J. Dai, K. He, and J. Sun. "Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 1635–1643 (see p. 128).
- [73] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra. "Embodied question answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018, pp. 2054–2063 (see pp. 126, 131, 148).
- [74] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra. "Visual dialog". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 326–335 (see pp. 96, 131).
- [75] A. J. Davison. "FutureMapping: The computational structure of spatial AI systems". In: *arXiv preprint arXiv:1803.11288* (2018) (see p. 59).
- [76] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. "MonoSLAM: Real-time single camera SLAM". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 6 (2007), pp. 1052–1067 (see pp. 56, 58).
- [77] E. Delage, H. Lee, and A. Y. Ng. "A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image". In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2. IEEE. 2006, pp. 2418–2428 (see p. 28).
- [78] O. Delalleau and Y. Bengio. "Shallow vs. deep sum-product networks". In: *Advances in Neural Information Processing Systems*. 2011, pp. 666–674 (see p. 14).
- [79] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 1–22 (see p. 85).

- [80] M. Denkowski and A. Lavie. “Meteor universal: Language specific translation evaluation for any target language”. In: *Proceedings of the ninth workshop on statistical machine translation*. 2014, pp. 376–380 (see p. 113).
- [81] H. Dhamo, N. Navab, and F. Tombari. “Object-Driven Multi-Layer Scene Decomposition From a Single Image”. In: *The IEEE International Conference on Computer Vision (ICCV)*. 2019 (see p. 31).
- [82] H. Dhamo, K. Tateno, I. Laina, N. Navab, and F. Tombari. “Peeking behind objects: Layered depth prediction from a single image”. In: *Pattern Recognition Letters* 125 (2019), pp. 333–340 (see p. 31).
- [83] H. Dhamo, K. Tateno, I. Laina, N. Navab, and F. Tombari. “Peeking behind objects: Layered depth prediction from a single image”. In: *Pattern Recognition Letters* 125 (2019), pp. 333–340 (see p. 155).
- [84] H. Dhamo, A. Farshad, I. Laina, N. Navab, G. D. Hager, F. Tombari, and C. Rupprecht. “Semantic Image Manipulation Using Scene Graphs”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 5213–5222 (see p. 126).
- [85] H. Dhamo, A. Farshad, I. Laina, N. Navab, G. D. Hager, F. Tombari, and C. Rupprecht. “Semantic Image Manipulation Using Scene Graphs”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 5213–5222 (see p. 155).
- [86] J. Donahue, P. Krähenbühl, and T. Darrell. “Adversarial feature learning”. In: *arXiv preprint arXiv:1605.09782* (2016) (see p. 97).
- [87] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. “Long-term recurrent convolutional networks for visual recognition and description”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 2625–2634 (see p. 98).
- [88] A. Dosovitskiy, J. Tobias Springenberg, and T. Brox. “Learning to Generate Chairs With Convolutional Neural Networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 1538–1546 (see p. 32, 34).
- [89] X. Du, T. Kurmann, P.-L. Chang, M. Allan, S. Ourselin, R. Sznitman, J. D. Kelly, and D. Stoyanov. “Articulated multi-instrument 2D pose estimation using fully convolutional networks”. In: *IEEE Transactions on Medical Imaging* (2018) (see p. 73).
- [90] J. Duchi, E. Hazan, and Y. Singer. “Adaptive subgradient methods for online learning and stochastic optimization”. In: *Journal of machine learning research* 12:Jul (2011), pp. 2121–2159 (see p. 17).
- [91] V. Dumoulin, J. Shlens, and M. Kudlur. “A Learned Representation For Artistic Style”. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings*. 2017 (see p. 130).
- [92] V. Dumoulin, E. Perez, N. Schucher, F. Strub, H. d. Vries, A. Courville, and Y. Bengio. “Feature-wise transformations”. In: *Distill* (2018). <https://distill.pub/2018/feature-wise-transformations> (see p. 130).
- [93] D. Eigen and R. Fergus. “Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture”. In: *Proc. Int. Conf. Computer Vision (ICCV)*. 2015 (see pp. 25, 29, 32, 44, 45, 49–52).
- [94] D. Eigen, C. Puhrsch, and R. Fergus. “Depth Map Prediction from a Single Image using a Multi-Scale Deep Network”. In: *Proc. Conf. Neural Information Processing Systems (NIPS)*. 2014 (see pp. 29, 32, 33, 37, 41, 44, 45).

- [95] A. El-Nouby, S. Sharma, H. Schulz, D. Hjelm, L. E. Asri, S. E. Kahou, Y. Bengio, and G. W. Taylor. "Tell, draw, and repeat: Generating and modifying images based on continual linguistic instruction". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 10304–10312 (see p. 126).
- [96] D. Elliott and F. Keller. "Image description using visual dependency representations". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 2013, pp. 1292–1302 (see p. 98).
- [97] J. Engel, T. Schöps, and D. Cremers. "LSD-SLAM: Large-scale direct monocular SLAM". In: *European conference on computer vision*. Springer. 2014, pp. 834–849 (see pp. 59–62, 64–66).
- [98] J. Engel, J. Sturm, and D. Cremers. "Semi-dense visual odometry for a monocular camera". In: *Proceedings of the IEEE international conference on computer vision*. 2013, pp. 1449–1456 (see pp. 59, 63, 64).
- [99] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*. <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html> (see pp. 135, 136, 140).
- [100] J. M. Facil, B. Ummenhofer, H. Zhou, L. Montesano, T. Brox, and J. Civera. "CAM-Convs: Camera-Aware Multi-Scale Convolutions for Single-View Depth". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 11826–11835 (see pp. 29, 47).
- [101] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler. "VSE++: Improving visual-semantic embeddings with hard negatives". In: *arXiv preprint arXiv:1707.05612* (2017) (see p. 101).
- [102] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, et al. "From captions to visual concepts and back". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1473–1482 (see p. 101).
- [103] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. "Every picture tells a story: Generating sentences from images". In: *European conference on computer vision*. Springer. 2010, pp. 15–29 (see pp. 96, 98).
- [104] O. Faugeras, Q.-T. Luong, and T. Papadopoulos. *The geometry of multiple images: the laws that govern the formation of multiple images of a scene and some of their applications*. MIT press, 2001 (see p. 28).
- [105] L. Fei-Fei, A. Iyer, C. Koch, and P. Perona. "What do we perceive in a glance of a real-world scene?" In: *Journal of vision* 7.1 (2007), pp. 10–10 (see p. 7).
- [106] Y. Feng, L. Ma, W. Liu, and J. Luo. "Unsupervised image captioning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4125–4134 (see pp. 100, 115, 116).
- [107] M. Firman, N. D. Campbell, L. Agapito, and G. J. Brostow. "Diversenet: When one right answer is not enough". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5598–5607 (see p. 84).
- [108] A. R. J. François, G. G. Medioni, and R. Waupotitsch. "Mirror Symmetry \Rightarrow 2-View Stereo Geometry". In: *Image and Vision Computing* (2003) (see p. 28).
- [109] D. Fried, R. Hu, V. Cirik, A. Rohrbach, J. Andreas, L.-P. Morency, T. Berg-Kirkpatrick, K. Saenko, D. Klein, and T. Darrell. "Speaker-follower models for vision-and-language navigation". In: *Advances in Neural Information Processing Systems*. 2018, pp. 3314–3325 (see pp. 125, 131).
- [110] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. "Devise: A deep visual-semantic embedding model". In: *Advances in neural information processing systems*. 2013, pp. 2121–2129 (see p. 101).

- [111] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. “Deep ordinal regression network for monocular depth estimation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2002–2011 (see p. 29).
- [112] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu. “Dual attention network for scene segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 3146–3154 (see p. 128).
- [113] D. Gálvez-López, M. Salas, J. D. Tardós, and J. Montiel. “Real-time monocular object slam”. In: *Robotics and Autonomous Systems* 75 (2016), pp. 435–449 (see p. 57).
- [114] C. Gan, Z. Gan, X. He, J. Gao, and L. Deng. “StyleNet: Generating attractive visual captions with styles”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 3137–3146 (see p. 99).
- [115] L. Garcia Peraza Herrera, W. Li, C. Gruijthuijsen, A. Devreker, G. Attilakos, J. Deprest, E. Vander Poorten, D. Stoyanov, T. Vercauteren, and S. Ourselin. “Real-Time Segmentation of Non-Rigid Surgical Tools based on Deep Learning and Tracking”. In: *CARE workshop at MICCAI*. Springer. 2016 (see p. 80).
- [116] R. Garg, V. K. BG, G. Carneiro, and I. Reid. “Unsupervised cnn for single view depth estimation: Geometry to the rescue”. In: *European Conference on Computer Vision*. Springer. 2016, pp. 740–756 (see p. 30).
- [117] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. “Vision meets Robotics: The KITTI Dataset”. In: *International Journal of Robotics Research (IJRR)* (2013) (see p. 40).
- [118] G. Ghazaei, I. Laina, C. Rupprecht, F. Tombari, N. Navab, and K. Nazarpour. “Dealing with ambiguity in robotic grasping via multiple predictions”. In: *Asian Conference on Computer Vision*. Springer. 2018, pp. 38–55 (see p. 72).
- [119] G. Ghazaei, I. Laina, C. Rupprecht, F. Tombari, N. Navab, and K. Nazarpour. “Dealing with ambiguity in robotic grasping via multiple predictions”. In: *Asian Conference on Computer Vision*. Springer. 2018, pp. 38–55 (see p. 155).
- [120] G. Ghiasi, H. Lee, M. Kudlur, V. Dumoulin, and J. Shlens. “Exploring the structure of a real-time, arbitrary neural artistic stylization network”. In: *British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017*. 2017 (see p. 130).
- [121] J. J. Gibson. “The perception of the visual world.” In: (1950) (see p. 26).
- [122] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal. “Explaining explanations: An overview of interpretability of machine learning”. In: *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*. IEEE. 2018, pp. 80–89 (see p. 6).
- [123] R. Girshick. “Fast R-CNN”. In: *The IEEE International Conference on Computer Vision (ICCV)*. 2015 (see p. 81).
- [124] R. Girshick, J. Donahue, T. Darrell, and J. Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587 (see p. 81).
- [125] X. Glorot, A. Bordes, and Y. Bengio. “Deep sparse rectifier neural networks”. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. 2011, pp. 315–323 (see p. 15).
- [126] C. Godard, O. Mac Aodha, and G. J. Brostow. “Unsupervised monocular depth estimation with left-right consistency”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 270–279 (see p. 30).

- [127] M. Goodale and A. Milner. "Separate visual pathways for perception and action." In: *Trends in neurosciences* 15.1 (1992), pp. 20–25 (see p. 7).
- [128] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. 2016 (see pp. 3, 13, 76).
- [129] L. Grady. "Random walks for image segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* 28.11 (2006), pp. 1768–1783 (see p. 128).
- [130] L. Grady, M.-P. Jolly, and A. Seitz. "Segmentation from a box". In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE. 2011, pp. 367–374 (see p. 128).
- [131] W. N. Greene, K. Ok, P. Lommel, and N. Roy. "Multi-level mapping: Real-time dense monocular SLAM". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 833–840 (see p. 59).
- [132] J. Gu, S. Joty, J. Cai, and G. Wang. "Unpaired Image Captioning by Language Pivoting". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 503–519 (see pp. 100, 115, 116).
- [133] J. Gu, S. Joty, J. Cai, H. Zhao, X. Yang, and G. Wang. "Unpaired Image Captioning via Scene Graph Alignments". In: *arXiv preprint arXiv:1903.10658* (2019) (see pp. 100, 101).
- [134] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. "Improved training of wasserstein gans". In: *Advances in Neural Information Processing Systems*. 2017, pp. 5767–5777 (see p. 110).
- [135] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. "Geodesic star convexity for interactive image segmentation". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE. 2010, pp. 3129–3136 (see p. 128).
- [136] D. Guo, F. Sun, H. Liu, T. Kong, B. Fang, and N. Xi. "A hybrid deep architecture for robotic grasp detection". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017 (see pp. 81, 82, 86, 88).
- [137] L. Guo, J. Liu, P. Yao, J. Li, and H. Lu. "MSCap: Multi-Style Image Captioning With Unpaired Stylized Text". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 (see p. 100).
- [138] X. Guo, H. Li, S. Yi, J. Ren, and X. Wang. "Learning monocular depth by distilling cross-domain stereo networks". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 484–500 (see pp. 29, 31).
- [139] A. Gupta, A. A. Efros, and M. Hebert. "Blocks world revisited: Image understanding using qualitative geometry and mechanics". In: *European Conference on Computer Vision*. Springer. 2010, pp. 482–496 (see p. 7).
- [140] A. Gupta, M. Hebert, T. Kanade, and D. M. Blei. "Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces". In: *Advances in neural information processing systems*. 2010, pp. 1288–1296 (see p. 28).
- [141] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik. "Cognitive mapping and planning for visual navigation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2616–2625 (see p. 60).
- [142] S. Gupta, P. Arbelaez, and J. Malik. "Perceptual organization and recognition of indoor scenes from RGB-D images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 564–571 (see pp. 49, 50).
- [143] D. Gurari, Q. Li, A. J. Stangl, A. Guo, C. Lin, K. Grauman, J. Luo, and J. P. Bigham. "VizWiz grand challenge: Answering visual questions from blind people". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3608–3617 (see p. 96).

- [144] A. Guzman-Rivera, P. Kohli, B. Glocker, J. Shotton, T. Sharp, A. Fitzgibbon, and S. Izadi. "Multi-output learning for camera relocalization". In: *Conference on Computer Vision and Pattern Recognition*. 2014 (see pp. 84, 85).
- [145] A. Guzman-Rivera, D. Batra, and P. Kohli. "Multiple choice learning: Learning to produce multiple structured outputs". In: *Advances in Neural Information Processing Systems*. 2012, pp. 1799–1807 (see p. 84).
- [146] M. A. K. Halliday. "Learning how to mean". In: *Foundations of language development*. Elsevier, 1975, pp. 239–265 (see p. 5).
- [147] A. Handa, T. Whelan, J. McDonald, and A. J. Davison. "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM". In: *2014 IEEE international conference on Robotics and automation (ICRA)*. IEEE. 2014, pp. 1524–1531 (see pp. 64, 65).
- [148] Z. Hao, Y. Li, S. You, and F. Lu. "Detail Preserving Depth Estimation from a Single Image Using Attention Guided Networks". In: *2018 International Conference on 3D Vision (3DV)*. IEEE. 2018, pp. 304–313 (see p. 29).
- [149] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003 (see p. 26).
- [150] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition". In: *arXiv preprint arXiv:1512.03385* (2015) (see pp. 19, 27, 33, 34).
- [151] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778 (see pp. 18, 19, 22, 34, 41, 42, 107, 112, 132, 140).
- [152] K. He, X. Zhang, S. Ren, and J. Sun. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1026–1034 (see p. 15).
- [153] R. Hebbalaguppe, K. McGuinness, J. Kuklyte, G. Healy, N. O'Connor, and A. Smeaton. "How interaction methods affect image segmentation: user experience in the task". In: *User-Centered Computer Vision (UCCV), 2013 1st IEEE Workshop on*. IEEE. 2013, pp. 19–24 (see p. 128).
- [154] V. Hedau, D. Hoiem, and D. Forsyth. "Recovering the spatial layout of cluttered rooms". In: *2009 IEEE 12th international conference on computer vision*. IEEE. 2009, pp. 1849–1856 (see p. 28).
- [155] J. F. Henriques and A. Vedaldi. "Mapnet: An allocentric spatial memory for mapping environments". In: *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8476–8484 (see p. 60).
- [156] F. Hill, K. Cho, and A. Korhonen. "Learning distributed representations of sentences from unlabelled data". In: *arXiv preprint arXiv:1602.03483* (2016) (see p. 102).
- [157] G. E. Hinton, S. Osindero, and Y.-W. Teh. "A fast learning algorithm for deep belief nets". In: *Neural computation* 18.7 (2006), pp. 1527–1554 (see p. 14).
- [158] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. "Improving neural networks by preventing co-adaptation of feature detectors". In: *arXiv preprint arXiv:1207.0580* (2012) (see p. 19).
- [159] S. Hochreiter and J. Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780 (see pp. 20, 104).
- [160] M. Hodosh, P. Young, and J. Hockenmaier. "Framing image description as a ranking task: Data, models and evaluation metrics". In: *Journal of Artificial Intelligence Research* 47 (2013), pp. 853–899 (see p. 98).

- [161] D. Hoiem, A. A. Efros, and M. Hebert. "Automatic photo pop-up". In: *ACM transactions on graphics (TOG)*. Vol. 24. 3. ACM. 2005, pp. 577–584 (see p. 28).
- [162] D. Hoiem, A. A. Efros, and M. Hebert. "Closing the loop in scene interpretation". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2008, pp. 1–8 (see p. 7).
- [163] D. Hoiem, A. Efros, M. Hebert, et al. "Geometric context from a single image". In: *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. Vol. 1. IEEE. 2005, pp. 654–661 (see pp. 26, 28).
- [164] D. Hoiem, A. N. Stein, A. A. Efros, and M. Hebert. "Recovering occlusion boundaries from a single image". In: *2007 IEEE 11th International Conference on Computer Vision*. IEEE. 2007, pp. 1–8 (see p. 28).
- [165] D. Hoiem, A. A. Efros, and M. Hebert. "Recovering surface layout from an image". In: *International Journal of Computer Vision* 75.1 (2007), pp. 151–172 (see p. 28).
- [166] B. K. Horn. "Obtaining shape from shading information". In: *The psychology of computer vision* (1975), pp. 115–155 (see p. 28).
- [167] J. Hu, L. Shen, and G. Sun. "Squeeze-and-excitation networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7132–7141 (see pp. 19, 130).
- [168] J. Hu, M. Ozay, Y. Zhang, and T. Okatani. "Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries". In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2019, pp. 1043–1051 (see p. 29).
- [169] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko. "Learning to reason: End-to-end module networks for visual question answering". In: *arXiv preprint arXiv:1704.05526* (2017) (see p. 131).
- [170] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. "Natural language object retrieval". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4555–4564 (see p. 131).
- [171] R. Hu, M. Rohrbach, and T. Darrell. "Segmentation from natural language expressions". In: *European Conference on Computer Vision*. Springer. 2016, pp. 108–124 (see p. 131).
- [172] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708 (see p. 19).
- [173] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. "Speed/accuracy trade-offs for modern convolutional object detectors". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7310–7311 (see p. 112).
- [174] P.-Y. Huang, F. Liu, S.-R. Shiang, J. Oh, and C. Dyer. "Attention-based multimodal neural machine translation". In: *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*. 2016, pp. 639–645 (see p. 96).
- [175] X. Huang and S. Belongie. "Arbitrary style transfer in real-time with adaptive instance normalization". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 1501–1510 (see p. 130).
- [176] D. Huk Park, L. Anne Hendricks, Z. Akata, A. Rohrbach, B. Schiele, T. Darrell, and M. Rohrbach. "Multimodal explanations: Justifying decisions and pointing to the evidence". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8779–8788 (see p. 96).
- [177] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne. "Imitation learning: A survey of learning methods". In: *ACM Computing Surveys (CSUR)* 50.2 (2017), pp. 1–35 (see p. 6).

- [178] E. Ilg, O. Cicek, S. Galesso, A. Klein, O. Makansi, F. Hutter, and T. Brox. "Uncertainty estimates and multi-hypotheses networks for optical flow". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 652–667 (see p. 84).
- [179] S. Ioffe and C. Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of The 32nd International Conference on Machine Learning*. 2015, pp. 448–456 (see pp. 19, 130).
- [180] M. Jaderberg, K. Simonyan, A. Zisserman, et al. "Spatial transformer networks". In: *Advances in neural information processing systems*. 2015, pp. 2017–2025 (see pp. 30, 59).
- [181] O. H. Jafari, O. Groth, A. Kirillov, M. Y. Yang, and C. Rother. "Analyzing modular cnn architectures for joint depth prediction and semantic segmentation". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 4620–4627 (see p. 30).
- [182] W.-D. Jang and C.-S. Kim. "Interactive Image Segmentation via Backpropagating Refinement Scheme". In: *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*. 2019 (see pp. 129, 135).
- [183] X. Ji, J. F. Henriques, and A. Vedaldi. "Invariant information clustering for unsupervised image classification and segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9865–9874 (see p. 97).
- [184] Y. Jiang, S. Moseson, and A. Saxena. "Efficient grasping from RGB-D images: Learning using a new rectangle representation". In: *International Conference on Robotics and Automation (ICRA)*. IEEE. 2011 (see pp. 81, 82, 87).
- [185] J. Jiao, Y. Cao, Y. Song, and R. Lau. "Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 53–69 (see p. 30).
- [186] J. Johnson, A. Karpathy, and L. Fei-Fei. "DenseCap: Fully convolutional localization networks for dense captioning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4565–4574 (see p. 98).
- [187] J. Johnson, B. Hariharan, L. van der Maaten, J. Hoffman, L. Fei-Fei, C. L. Zitnick, and R. Girshick. "Inferring and Executing Programs for Visual Reasoning". In: *arXiv preprint arXiv:1705.03633* (2017) (see p. 131).
- [188] A. Karpathy and L. Fei-Fei. "Deep visual-semantic alignments for generating image descriptions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3128–3137 (see pp. 98, 113, 114, 116).
- [189] A. Karpathy, A. Joulin, and L. F. Fei-Fei. "Deep fragment embeddings for bidirectional image sentence mapping". In: *Advances in neural information processing systems*. 2014, pp. 1889–1897 (see pp. 5, 101).
- [190] K. Karsch, C. Liu, and S. B. Kang. "Depth extraction from video using non-parametric sampling". In: *Proc. Europ. Conf. Computer Vision (ECCV)*. 2012, pp. 775–788 (see pp. 28, 45, 46).
- [191] M. Kass, A. Witkin, and D. Terzopoulos. "Snakes: Active contour models". In: *International journal of computer vision* 1.4 (1988), pp. 321–331 (see p. 128).
- [192] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg. "Referitgame: Referring to objects in photographs of natural scenes". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 787–798 (see p. 131).
- [193] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg. "A survey of research on cloud robotics and automation." In: () (see p. 81).

- [194] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb. "Real-time 3d reconstruction in dynamic scenes using point-based fusion". In: *2013 International Conference on 3D Vision-3DV 2013*. IEEE. 2013, pp. 1–8 (see pp. 48, 66).
- [195] I. Kemelmacher-Shlizerman and R. Basri. "3D face reconstruction from a single image using a single reference face shape". In: *IEEE transactions on pattern analysis and machine intelligence* 33.2 (2010), pp. 394–405 (see p. 26).
- [196] A. Kendall and Y. Gal. "What uncertainties do we need in bayesian deep learning for computer vision?" In: *Advances in neural information processing systems*. 2017, pp. 5574–5584 (see pp. 29, 62).
- [197] A. Kendall, Y. Gal, and R. Cipolla. "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7482–7491 (see p. 30).
- [198] C. Kerl, J. Sturm, and D. Cremers. "Robust Odometry Estimation for RGB-D Cameras". In: *Proc. Int. Conf. on Robotics and Automation (ICRA)*. 2013 (see p. 48).
- [199] K. Khoshelham and S. O. Elberink. "Accuracy and resolution of kinect depth data for indoor mapping applications". In: *Sensors* 12.2 (2012), pp. 1437–1454 (see p. 36).
- [200] J.-H. Kim, S.-W. Lee, D. Kwak, M.-O. Heo, J. Kim, J.-W. Ha, and B.-T. Zhang. "Multimodal residual learning for visual qa". In: *Advances in neural information processing systems*. 2016, pp. 361–369 (see p. 101).
- [201] S. Kim, K. Park, K. Sohn, and S. Lin. "Unified depth prediction and intrinsic image decomposition from a single image via joint convolutional neural fields". In: *European conference on computer vision*. Springer. 2016, pp. 143–159 (see p. 30).
- [202] D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014) (see pp. 17, 112).
- [203] D. P. Kingma and M. Welling. "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. 2014 (see p. 60).
- [204] R. Kiros, R. Salakhutdinov, and R. Zemel. "Multimodal neural language models". In: *International Conference on Machine Learning*. 2014, pp. 595–603 (see p. 101).
- [205] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler. "Skip-thought vectors". In: *Advances in neural information processing systems*. 2015, pp. 3294–3302 (see p. 102).
- [206] R. Kiros, R. Salakhutdinov, and R. S. Zemel. "Unifying visual-semantic embeddings with multimodal neural language models". In: *arXiv preprint arXiv:1411.2539* (2014) (see p. 101).
- [207] G. Klein and D. Murray. "Parallel tracking and mapping for small AR workspaces". In: *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society. 2007, pp. 1–10 (see pp. 58, 60).
- [208] M. Klodt and A. Vedaldi. "Supervising the new with the old: learning SFM from SFM". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 698–713 (see p. 31).
- [209] P. Koehn. *Statistical machine translation*. Cambridge University Press, 2009 (see p. 115).
- [210] A. Kolesnikov and C. H. Lampert. "Seed, expand and constrain: Three principles for weakly-supervised image segmentation". In: *European Conference on Computer Vision*. Springer. 2016, pp. 695–711 (see p. 128).

- [211] J. Konrad, M. Wang, and P. Ishwar. “2d-to-3d image conversion by learning depth from examples”. In: *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE. 2012, pp. 16–22 (see p. 29).
- [212] T. Kontogianni, M. Gygli, J. Uijlings, and V. Ferrari. “Continuous adaptation for interactive object segmentation by learning from corrections”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 579–596 (see pp. 129, 135).
- [213] S. Kornblith, J. Shlens, and Q. V. Le. “Do better imagenet models transfer better?”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2661–2671 (see p. 43).
- [214] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. “Visual genome: Connecting language and vision using crowdsourced dense image annotations”. In: *International Journal of Computer Vision* 123.1 (2017), pp. 32–73 (see pp. 103, 111).
- [215] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105 (see pp. 14, 19, 22, 32, 33, 41, 42).
- [216] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. “Babytalk: Understanding and generating simple image descriptions”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.12 (2013), pp. 2891–2903 (see pp. 96, 98, 100).
- [217] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. “g 2 o: A general framework for graph optimization”. In: *2011 IEEE International Conference on Robotics and Automation*. IEEE. 2011, pp. 3607–3613 (see pp. 57, 63).
- [218] S. Kumra and C. Kanan. “Robotic grasp detection using deep convolutional neural networks”. In: *arXiv preprint arXiv:1611.08036* (2016) (see pp. 81, 82, 86, 88).
- [219] T. Kurmann, P. M. Neila, X. Du, P. Fua, D. Stoyanov, S. Wolf, and R. Sznitman. “Simultaneous recognition and pose estimation of instruments in minimally invasive surgery”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2017, pp. 505–513 (see p. 73).
- [220] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger. “From word embeddings to document distances”. In: *International Conference on Machine Learning*. 2015, pp. 957–966 (see p. 113).
- [221] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Mallocci, A. Kolesnikov, et al. “The open images dataset v4”. In: *International Journal of Computer Vision* (2020), pp. 1–26 (see pp. 107, 112, 116).
- [222] P. Kuznetsova, V. Ordonez, A. C. Berg, T. L. Berg, and Y. Choi. “Collective generation of natural image descriptions”. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics. 2012, pp. 359–368 (see p. 98).
- [223] P. Kuznetsova, V. Ordonez, T. L. Berg, and Y. Choi. “Treetalk: Composition and compression of trees for image descriptions”. In: *Transactions of the Association for Computational Linguistics* 2 (2014), pp. 351–362 (see p. 98).
- [224] Y. Kuznetsov, J. Stuckler, and B. Leibe. “Semi-supervised deep learning for monocular depth map prediction”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6647–6655 (see pp. 30, 31).
- [225] L. Ladicky, J. Shi, and M. Pollefeys. “Pulling Things out of Perspective”. In: *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2014, pp. 89–96 (see pp. 28, 45).

- [226] I. Laina, N. Rieke, C. Rupprecht, J. P. Vizcaíno, A. Eslami, F. Tombari, and N. Navab. “Concurrent segmentation and localization for tracking of surgical instruments”. In: *International conference on medical image computing and computer-assisted intervention*. Springer. 2017, pp. 664–672 (see p. 72).
- [227] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. “Deeper depth prediction with fully convolutional residual networks”. In: *2016 Fourth international conference on 3D vision (3DV)*. IEEE. 2016, pp. 239–248 (see pp. 25, 26, 66, 151).
- [228] I. Laina, C. Rupprecht, and N. Navab. “Towards Unsupervised Image Captioning with Shared Multimodal Embeddings”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 7414–7424 (see pp. 5, 97).
- [229] G. Lample, M. Ott, A. Conneau, L. Denoyer, and M. Ranzato. “Phrase-Based & Neural Unsupervised Machine Translation”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2018 (see p. 101).
- [230] G. Lample, A. Conneau, L. Denoyer, and M. Ranzato. “Unsupervised machine translation using monolingual corpora only”. In: *International Conference on Learning Representations (ICLR)*. 2018 (see pp. 97, 101).
- [231] K. Lasinger, R. Ranftl, K. Schindler, and V. Koltun. “Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer”. In: *arXiv preprint arXiv:1907.01341* (2019) (see pp. 30, 47).
- [232] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324 (see p. 17).
- [233] S. Lee, S. P. S. Prakash, M. Cogswell, V. Ranjan, D. Crandall, and D. Batra. “Stochastic multiple choice learning for training diverse deep ensembles”. In: *Advances in Neural Information Processing Systems*. 2016 (see p. 84).
- [234] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. “Image segmentation with a bounding box prior”. In: *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE. 2009, pp. 277–284 (see p. 128).
- [235] I. Lenz, H. Lee, and A. Saxena. “Deep learning for detecting robotic grasps”. In: *The International Journal of Robotics Research* (2015) (see pp. 81–83, 85–88).
- [236] A. Levin, D. Lischinski, and Y. Weiss. “Colorization using optimization”. In: *ACM transactions on graphics (tog)*. Vol. 23. 3. ACM. 2004, pp. 689–694 (see p. 40).
- [237] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen. “Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection”. In: *arXiv preprint arXiv:1603.02199* (2016) (see p. 81).
- [238] B. Li, C. Shen, Y. Dai, A. V. den Hengel, and M. He. “Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs”. In: *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1119–1127 (see pp. 29, 45, 46).
- [239] D. Li, X. He, Q. Huang, M.-T. Sun, and L. Zhang. “Generating diverse and accurate visual captions by comparative adversarial learning”. In: *arXiv preprint arXiv:1804.00861* (2018) (see p. 99).
- [240] J. Li, R. Klein, and A. Yao. “A two-streamed network for estimating fine-scaled depth maps from single rgb images”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 3372–3380 (see p. 29).
- [241] L.-J. Li, R. Socher, and L. Fei-Fei. “Towards total scene understanding: Classification, annotation and segmentation in an automatic framework”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2009, pp. 2036–2043 (see p. 7).

- [242] Q. Li, Q. Tao, S. Joty, J. Cai, and J. Luo. "Vqa-e: Explaining, elaborating, and enhancing your answers for visual questions". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 552–567 (see p. 96).
- [243] R. Li, K. Xian, C. Shen, Z. Cao, H. Lu, and L. Hang. "Deep attention-based classification network for robust depth prediction". In: *Asian Conference on Computer Vision*. Springer. 2018, pp. 663–678 (see pp. 29, 47).
- [244] R. Li, S. Wang, Z. Long, and D. Gu. "Undeepvo: Monocular visual odometry through unsupervised deep learning". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 7286–7291 (see pp. 30, 59).
- [245] S. Li, G. Kulkarni, T. L. Berg, A. C. Berg, and Y. Choi. "Composing simple image descriptions using web-scale n-grams". In: *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics. 2011, pp. 220–228 (see p. 98).
- [246] Z. Li and N. Snavely. "Megadepth: Learning single-view depth prediction from internet photos". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018 (see p. 29).
- [247] Z. Li, Q. Chen, and V. Koltun. "Interactive Image Segmentation With Latent Diversity". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 (see pp. 84, 85).
- [248] Z. Li, Q. Chen, and V. Koltun. "Interactive image segmentation with latent diversity". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 577–585 (see p. 129).
- [249] J. Liew, Y. Wei, W. Xiong, S.-H. Ong, and J. Feng. "Regional interactive image segmentation networks". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE. 2017, pp. 2746–2754 (see p. 129).
- [250] C.-Y. Lin. "Rouge: A package for automatic evaluation of summaries". In: *Text Summarization Branches Out* (2004) (see p. 113).
- [251] D. Lin, J. Dai, J. Jia, K. He, and J. Sun. "Scribblesup: Scribble-supervised convolutional networks for semantic segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 3159–3167 (see p. 128).
- [252] G. Lin, A. Milan, C. Shen, and I. Reid. "Refinenet: Multi-path refinement networks with identity mappings for high-resolution semantic segmentation". In: *arXiv preprint arXiv:1611.06612* (2016) (see p. 128).
- [253] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755 (see pp. 97, 140).
- [254] H. Ling and S. Fidler. "Teaching machines to describe images via natural language feedback". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Curran Associates Inc. 2017, pp. 5075–5085 (see p. 96).
- [255] B. Liu, S. Gould, and D. Koller. "Single image depth estimation from predicted semantic labels". In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE. 2010, pp. 1253–1260 (see p. 28).
- [256] C. Liu, J. Yuen, and A. Torralba. "Sift flow: Dense correspondence across scenes and its applications". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 33.5 (2011), pp. 978–994 (see p. 29).

- [257] F. Liu, C. Shen, and G. Lin. “Deep Convolutional Neural Fields for Depth Estimation from a Single Image”. In: *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 5162–5170 (see pp. [29](#), [45](#), [46](#)).
- [258] M. Liu, M. Salzmann, and X. He. “Discrete-continuous depth estimation from a single image”. In: *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2014, pp. 716–723 (see pp. [29](#), [41](#), [45](#), [46](#)).
- [259] X. Liu, A. Sinha, M. Ishii, G. D. Hager, A. Reiter, R. H. Taylor, and M. Unberath. “Dense Depth Estimation in Monocular Endoscopy with Self-supervised Learning Methods”. In: *IEEE transactions on medical imaging* (2019) (see p. [31](#)).
- [260] J. Long, E. Shelhamer, and T. Darrell. “Fully convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 3431–3440 (see pp. [32](#), [49](#), [50](#), [128](#), [135](#), [136](#)).
- [261] D. G. Lowe. “Distinctive image features from scale-invariant keypoints”. In: *International journal of computer vision* 60.2 (2004), pp. 91–110 (see pp. [58](#), [103](#)).
- [262] J. Lu, A. Kannan, J. Yang, D. Parikh, and D. Batra. “Best of both worlds: Transferring knowledge from discriminative learning to a generative visual dialog model”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 314–324 (see p. [131](#)).
- [263] J. Lu, J. Yang, D. Batra, and D. Parikh. “Hierarchical question-image co-attention for visual question answering”. In: *Advances In Neural Information Processing Systems*. 2016, pp. 289–297 (see p. [131](#)).
- [264] J. Lu, C. Xiong, D. Parikh, and R. Socher. “Knowing when to look: Adaptive attention via a visual sentinel for image captioning”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 375–383 (see p. [99](#)).
- [265] J. Lu, J. Yang, D. Batra, and D. Parikh. “Neural baby talk”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7219–7228 (see pp. [100](#), [103](#), [123](#)).
- [266] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, and A. Yuille. “Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding”. In: *IEEE transactions on pattern analysis and machine intelligence* 42.10 (2019), pp. 2624–2641 (see p. [30](#)).
- [267] R. Luo and G. Shakhnarovich. “Comprehension-guided referring expressions”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 7102–7111 (see p. [131](#)).
- [268] Y. Luo, J. Ren, M. Lin, J. Pang, W. Sun, H. Li, and L. Lin. “Single view stereo matching”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 155–163 (see p. [31](#)).
- [269] F. Ma, G. V. Cavalheiro, and S. Karaman. “Self-supervised Sparse-to-Dense: Self-supervised Depth Completion from LiDAR and Monocular Camera”. In: (2019) (see p. [31](#)).
- [270] A. L. Maas, A. Y. Hannun, and A. Y. Ng. “Rectifier nonlinearities improve neural network acoustic models”. In: *Proc. icml*. Vol. 30. 1. 2013, p. 3 (see p. [15](#)).
- [271] L. v. d. Maaten and G. Hinton. “Visualizing data using t-SNE”. In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605 (see pp. [119](#), [146](#)).
- [272] S. Mahadevan, P. Voigtlaender, and B. Leibe. “Iteratively trained interactive segmentation”. In: *arXiv preprint arXiv:1805.04398* (2018) (see p. [129](#)).
- [273] R. Mahjourian, M. Wicke, and A. Angelova. “Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5667–5675 (see p. [30](#)).

- [274] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg. “Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics”. In: *arXiv preprint arXiv:1703.09312* (2017) (see p. 81).
- [275] F. Mahmood and N. J. Durr. “Deep learning and conditional random fields-based depth estimation and topographical reconstruction from conventional endoscopy”. In: *Medical image analysis* 48 (2018), pp. 230–243 (see p. 31).
- [276] S. Majumder and A. Yao. “Content-Aware Multi-Level Guidance for Interactive Instance Segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 11602–11611 (see p. 129).
- [277] M. Malinowski, M. Rohrbach, and M. Fritz. “Ask your neurons: A deep learning approach to visual question answering”. In: *International Journal of Computer Vision* 125.1-3 (2017), pp. 110–135 (see p. 131).
- [278] M. Malinowski, M. Rohrbach, and M. Fritz. “Ask your neurons: A neural-based approach to answering questions about images”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1–9 (see p. 131).
- [279] F. Manhardt, D. M. Arroyo, C. Rupprecht, B. Busam, T. Birdal, N. Navab, and F. Tombari. “Explaining the Ambiguity of Object Detection and 6D Pose from Visual Data”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 6841–6850 (see p. 84).
- [280] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool. “Deep extreme cut: From extreme points to object segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 616–625 (see p. 129).
- [281] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky. “The Stanford CoreNLP natural language processing toolkit”. In: *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. 2014, pp. 55–60 (see pp. 103, 111).
- [282] J. Mao, X. Wei, Y. Yang, J. Wang, Z. Huang, and A. L. Yuille. “Learning like a child: Fast novel visual concept learning from sentence descriptions of images”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2533–2541 (see p. 99).
- [283] A. Mathews, L. Xie, and X. He. “SemStyle: Learning to generate stylised image captions using unaligned text”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8591–8600 (see pp. 99, 100).
- [284] A. P. Mathews, L. Xie, and X. He. “Senticap: Generating image descriptions with sentiments”. In: *Thirtieth AAAI conference on artificial intelligence*. 2016 (see p. 99).
- [285] P. Mazzone, R. A. Andersen, and M. I. Jordan. “A more biologically plausible learning rule for neural networks.” In: *Proceedings of the National Academy of Sciences* 88.10 (1991), pp. 4433–4437 (see p. 14).
- [286] J. McCormac, A. Handa, A. Davison, and S. Leutenegger. “Semanticfusion: Dense 3d semantic mapping with convolutional neural networks”. In: *2017 IEEE International Conference on Robotics and automation (ICRA)*. IEEE. 2017, pp. 4628–4635 (see p. 60).
- [287] W. S. McCulloch and W. Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133 (see p. 14).
- [288] G. McLachlan and D. Peel. *Finite mixture models*. John Wiley & Sons, 2004 (see p. 85).
- [289] R. Memisevic and C. Conrad. “Stereopsis via deep learning”. In: *NIPS Workshop on Deep Learning*. Vol. 1. 2011 (see p. 27).
- [290] A. T. Miller and P. K. Allen. “Graspit! a versatile simulator for robotic grasping”. In: *IEEE Robotics & Automation Magazine* (2004) (see p. 81).

- [291] A. Miller, S. Knoop, H. Christensen, and P. Allen. "Automatic grasp planning using shape primitives". In: *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*. IEEE (see p. 81).
- [292] G. A. Miller. "WordNet: a lexical database for English". In: *Communications of the ACM* 38.11 (1995), pp. 39–41 (see pp. 103, 108).
- [293] M. Minsky and S. A. Papert. *Perceptrons: an introduction to computational geometry*. 1969 (see p. 14).
- [294] I. Misra, R. Girshick, R. Fergus, M. Hebert, A. Gupta, and L. Van Der Maaten. "Learning by asking questions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 11–20 (see pp. 5, 148).
- [295] T. Miyato and M. Koyama. "cGANs with Projection Discriminator". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. 2018 (see p. 130).
- [296] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. "Spectral normalization for generative adversarial networks". In: *arXiv preprint arXiv:1802.05957* (2018) (see p. 19).
- [297] A. Mnih and G. Hinton. "Three new graphical models for statistical language modelling". In: *Proceedings of the 24th international conference on Machine learning*. ACM. 2007, pp. 641–648 (see p. 101).
- [298] A. Mousavian, H. Pirsiavash, and J. Koščeká. "Joint semantic segmentation and depth estimation with deep convolutional networks". In: *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE. 2016, pp. 611–619 (see pp. 29, 30, 50, 52).
- [299] D. P. Mukherjee, A. Zisserman, and J. M. Brady. "Shape from Symmetry – Detecting and Exploiting Symmetry in Affine Images". In: *Philosophical Transactions of the Royal Society of London* 351 (1995), pp. 77–106 (see p. 28).
- [300] D. P. Mukherjee, A. P. Zisserman, M. Brady, and F. Smith. "Shape from symmetry: Detecting and exploiting symmetry in affine images". In: *Philosophical Transactions of the Royal Society of London. Series A: Physical and Engineering Sciences* 351.1695 (1995), pp. 77–106 (see p. 26).
- [301] T. Munkhdalai, X. Yuan, S. Mehri, and A. Trischler. "Rapid adaptation with conditionally shifted neurons". In: *arXiv preprint arXiv:1712.09926* (2017) (see p. 130).
- [302] R. Mur-Artal and J. D. Tardós. "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras". In: *IEEE Transactions on Robotics* 33.5 (2017), pp. 1255–1262 (see pp. 56, 58).
- [303] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. "ORB-SLAM: a versatile and accurate monocular SLAM system". In: *IEEE transactions on robotics* 31.5 (2015), pp. 1147–1163 (see pp. 58, 60, 65, 66).
- [304] V. Nair and G. E. Hinton. "Rectified linear units improve restricted boltzmann machines". In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, pp. 807–814 (see p. 15).
- [305] J. Nath Kundu, P. Krishna Uppala, A. Pahuja, and R Venkatesh Babu. "Adadepth: Unsupervised content congruent adaptation for depth estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2656–2665 (see p. 29).
- [306] P. K. Nathan Silberman Derek Hoiem and R. Fergus. "Indoor Segmentation and Support Inference from RGBD Images". In: *ECCV*. 2012 (see pp. 25, 39, 65).
- [307] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. "DTAM: Dense tracking and mapping in real-time". In: *2011 international conference on computer vision*. IEEE. 2011, pp. 2320–2327 (see pp. 58, 59).

- [308] S. Niklaus, L. Mai, J. Yang, and F. Liu. “3D Ken Burns effect from a single image”. In: *ACM Transactions on Graphics (TOG)* 38.6 (2019), p. 184 (see p. 31).
- [309] A. Oliva and A. Torralba. “Modeling the shape of the scene: A holistic representation of the spatial envelope”. In: *Int. Journal of Computer Vision (IJCV)* (2014), pp. 145–175 (see p. 28).
- [310] A. Oliva. “Gist of the scene”. In: *Neurobiology of attention*. Elsevier, 2005, pp. 251–256 (see p. 7).
- [311] V. Ordonez, G. Kulkarni, and T. L. Berg. “Im2text: Describing images using 1 million captioned photographs”. In: *Advances in neural information processing systems*. 2011, pp. 1143–1151 (see p. 98).
- [312] A. B. Owen. “A robust hybrid of lasso and ridge regression”. In: *Contemporary Mathematics* 443 (2007), pp. 59–72 (see p. 37).
- [313] D. Pakhomov, V. Premachandran, M. Allan, M. Azizian, and N. Navab. “Deep residual learning for instrument segmentation in robotic surgery”. In: *arXiv preprint arXiv:1703.08580* (2017) (see p. 80).
- [314] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui. “Jointly modeling embedding and translation to bridge video and language”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 4594–4602 (see p. 101).
- [315] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu. “Semantic image synthesis with spatially-adaptive normalization”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2337–2346 (see p. 130).
- [316] D. Pathak, E. Shelhamer, J. Long, and T. Darrell. “Fully convolutional multi-class multiple instance learning”. In: *arXiv preprint arXiv:1412.7144* (2014) (see p. 128).
- [317] J. Pennington, R. Socher, and C. Manning. “Glove: Global vectors for word representation”. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543 (see pp. 112, 140).
- [318] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. “Film: Visual reasoning with a general conditioning layer”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018 (see pp. 129, 130, 132, 140, 141).
- [319] A. Pilzer, S. Lathuiliere, N. Sebe, and E. Ricci. “Refine and distill: Exploiting cycle-inconsistency and knowledge distillation for unsupervised monocular depth estimation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9768–9777 (see p. 30).
- [320] A. Pilzer, D. Xu, M. Puscas, E. Ricci, and N. Sebe. “Unsupervised adversarial depth estimation using cycled generative networks”. In: *2018 International Conference on 3D Vision (3DV)*. IEEE. 2018, pp. 587–595 (see p. 30).
- [321] M. Pizzoli, C. Forster, and D. Scaramuzza. “REMODE: Probabilistic, monocular dense reconstruction in real time”. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2014, pp. 2609–2616 (see pp. 65, 66).
- [322] M. Poggi, F. Tosi, and S. Mattoccia. “Learning monocular depth estimation with unsupervised trinocular assumptions”. In: *2018 International Conference on 3D Vision (3DV)*. IEEE. 2018, pp. 324–333 (see p. 30).
- [323] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe. “Full-Resolution Residual Networks for Semantic Segmentation in Street Scenes”. In: *arXiv preprint arXiv:1611.08323* (2016) (see p. 128).
- [324] M. C. Potter. “Short-term conceptual memory for pictures.” In: *Journal of experimental psychology: human learning and memory* 2.5 (1976), p. 509 (see p. 7).

- [325] M. C. Potter and E. I. Levy. "Recognition memory for a rapid sequence of pictures." In: *Journal of experimental psychology* 81.1 (1969), p. 10 (see p. 7).
- [326] B. L. Price, B. Morse, and S. Cohen. "Geodesic graph cut for interactive image segmentation". In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE. 2010, pp. 3161–3168 (see p. 128).
- [327] T. Probst, K.-K. Maninis, A. Chhatkuli, M. Ourak, E. Vander Poorten, and L. Van Gool. "Automatic tool landmark detection for stereo vision in robot-assisted retinal surgery". In: *IEEE Robotics and Automation Letters* 3.1 (2017), pp. 612–619 (see p. 73).
- [328] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba. "Virtualhome: Simulating household activities via programs". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8494–8502 (see p. 125).
- [329] X. Qi, Z. Liu, J. Shi, H. Zhao, and J. Jia. "Augmented feedback in semantic segmentation under image level supervision". In: *European Conference on Computer Vision*. Springer. 2016, pp. 90–105 (see p. 128).
- [330] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia. "Geonet: Geometric neural network for joint depth and surface normal estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 283–291 (see p. 30).
- [331] N. Qian. "On the momentum term in gradient descent learning algorithms". In: *Neural networks* 12.1 (1999), pp. 145–151 (see p. 17).
- [332] R. Qiao, L. Liu, C. Shen, and A. Van Den Hengel. "Less is more: zero-shot learning from online textual documents with noise suppression". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2249–2257 (see p. 96).
- [333] T. Qiao, J. Zhang, D. Xu, and D. Tao. "MirrorGAN: Learning Text-to-image Generation by Re-description". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1505–1514 (see p. 126).
- [334] M. Rajchl, M. C. Lee, O. Oktay, K. Kamnitsas, J. Passerat-Palmbach, W. Bai, M. Damodaram, M. A. Rutherford, J. V. Hajnal, B. Kainz, et al. "Deepcut: Object segmentation from bounding box annotations using convolutional neural networks". In: *IEEE transactions on medical imaging* 36.2 (2017), pp. 674–683 (see p. 128).
- [335] K. Rakelly, E. Shelhamer, T. Darrell, A. A. Efros, and S. Levine. "Few-shot segmentation propagation with guided networks". In: *arXiv preprint arXiv:1806.07373* (2018) (see p. 130).
- [336] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black. "Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12240–12249 (see p. 30).
- [337] J. Redmon and A. Angelova. "Real-time grasp detection using convolutional neural networks". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015 (see pp. 81, 82, 84, 86, 88).
- [338] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. "You only look once: Unified, real-time object detection". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (see p. 81).
- [339] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. "Generative adversarial text to image synthesis". In: *International Conference on Machine Learning*. PMLR. 2016, pp. 1060–1069 (see p. 126).

- [340] A. Reiter, P. K. Allen, and T. Zhao. "Feature classification for tracking articulated surgical tools". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2012, pp. 592–600 (see p. 73).
- [341] A. Reiter, P. K. Allen, and T. Zhao. "Marker-less articulated surgical tool detection". In: *Proc. Computer assisted radiology and surgery*. Vol. 7. 2012, pp. 175–176 (see p. 73).
- [342] S. Ren, K. He, R. Girshick, and J. Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems*. 2015, pp. 91–99 (see pp. 81, 84, 89, 107, 112).
- [343] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. "Self-critical sequence training for image captioning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 7008–7024 (see pp. 99, 122).
- [344] N. Rieke, D. J. Tan, C. Amat di San Filippo, F. Tombari, M. Alsheakhali, V. Belagiannis, A. Eslami, and N. Navab. "Real-time Localization of Articulated Surgical Instruments in Retinal Microsurgery". In: *Medical Image Analysis* 34 (2016) (see pp. 73, 77, 79).
- [345] N. Rieke, D. J. Tan, F. Tombari, J. P. Vizcaíno, C. A. di San Filippo, A. Eslami, and N. Navab. "Real-Time Online Adaption for Robust Instrument Tracking and Pose Estimation". In: *MICCAI*. Springer. 2016, pp. 422–430 (see pp. 73, 78, 79).
- [346] M. Rochan, L. Ye, and Y. Wang. "Video Summarization Using Fully Convolutional Sequence Networks". In: *arXiv preprint arXiv:1805.10538* (2018) (see p. 90).
- [347] O. Ronneberger, P. Fischer, and T. Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *MICCAI*. Springer. 2015, pp. 234–241 (see pp. 52, 76, 78, 79).
- [348] F. Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6 (1958), p. 386 (see p. 14).
- [349] C. Rother, V. Kolmogorov, and A. Blake. "Grabcut: Interactive foreground extraction using iterated graph cuts". In: *ACM transactions on graphics (TOG)*. Vol. 23. 3. ACM. 2004, pp. 309–314 (see p. 128).
- [350] A. Roy and S. Todorovic. "Monocular Depth Estimation Using Neural Regression Forest". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR (CVPR)*. 2016 (see pp. 29, 45).
- [351] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. "ORB: An efficient alternative to SIFT or SURF". In: *2011 International Conference on Computer Vision*. IEEE. 2011, pp. 2564–2571 (see p. 58).
- [352] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning representations by back-propagating errors". In: *nature* 323.6088 (1986), p. 533 (see p. 14).
- [353] C. Rupprecht, E. Huaroc, M. Baust, and N. Navab. "Deep active contours". In: *arXiv preprint arXiv:1607.05074* (2016) (see p. 128).
- [354] C. Rupprecht, I. Laina, N. Navab, G. D. Hager, and F. Tombari. "Guide me: Interacting with deep networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8551–8561 (see p. 125).
- [355] C. Rupprecht*, I. Laina*, N. Navab, G. D. Hager, and F. Tombari. "Guide me: Interacting with deep networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8551–8561 (see p. 155).
- [356] C. Rupprecht, L. Peter, and N. Navab. "Image segmentation in twenty questions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 3314–3322 (see pp. 128, 135).

- [357] C. Rupprecht, I. Laina, R. DiPietro, M. Baust, F. Tombari, N. Navab, and G. D. Hager. "Learning in an uncertain world: Representing ambiguity through multiple hypotheses". In: *International Conference on Computer Vision (ICCV)*. 2017 (see pp. 62, 84).
- [358] C. Rupprecht, I. Laina, R. DiPietro, M. Baust, F. Tombari, N. Navab, and G. D. Hager. "Learning in an uncertain world: Representing ambiguity through multiple hypotheses". In: *International Conference on Computer Vision (ICCV)*. 2017 (see p. 155).
- [359] O. Russakovsky, L.-J. Li, and L. Fei-Fei. "Best of both worlds: human-machine collaboration for object annotation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 2121–2131 (see p. 129).
- [360] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. "ImageNet large scale visual recognition challenge". In: *International journal of computer vision* 115.3 (2015), pp. 211–252 (see pp. 22, 34, 41, 99, 107).
- [361] S. J. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited, 2016 (see p. 3).
- [362] T. Sakinis, F. Milletari, H. Roth, P. Korfiatis, P. Kostandy, K. Philbrick, Z. Akkus, Z. Xu, D. Xu, and B. J. Erickson. "Interactive segmentation of medical images through fully convolutional neural networks". In: *arXiv preprint arXiv:1903.08205* (2019) (see p. 129).
- [363] T. Salimans and D. P. Kingma. "Weight normalization: A simple reparameterization to accelerate training of deep neural networks". In: *Advances in neural information processing systems*. 2016, pp. 901–909 (see p. 19).
- [364] D. Sarikaya, J. Corso, and K. Guru. "Detection and localization of robotic tools in robot-assisted surgery videos using deep neural networks for region proposal and detection". In: *IEEE Transactions on Medical Imaging* (2017) (see p. 73).
- [365] A. Saxena, M. Sun, and A. Ng. "Make3d: Learning 3d scene structure from a single still image". In: *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)* 12.5 (2009), pp. 824–840 (see pp. 28, 39, 40).
- [366] A. Saxena, S. H. Chung, and A. Y. Ng. "Learning depth from single monocular images". In: *Advances in Neural Information Processing Systems*. 2005, pp. 1161–1168 (see pp. 26, 28).
- [367] A. Saxena, J. Driemeyer, and A. Y. Ng. "Robotic grasping of novel objects using vision". In: *The International Journal of Robotics Research* (2008) (see p. 81).
- [368] D. Scharstein and R. Szeliski. "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms". In: *International journal of computer vision* 47.1-3 (2002), pp. 7–42 (see p. 27).
- [369] P. G. Schyns and A. Oliva. "From blobs to boundary edges: Evidence for time-and spatial-scale-dependent scene recognition". In: *Psychological science* 5.4 (1994), pp. 195–200 (see p. 7).
- [370] B. Settles. *Active learning literature survey*. Tech. rep. University of Wisconsin-Madison Department of Computer Sciences, 2009 (see p. 148).
- [371] P. Sharma, N. Ding, S. Goodman, and R. Soricut. "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1. 2018, pp. 2556–2565 (see p. 111).
- [372] S. Sharma, D. Suhubdy, V. Michalski, S. E. Kahou, and Y. Bengio. "Chatpainter: Improving text to image generation using dialogue". In: *arXiv preprint arXiv:1802.08216* (2018) (see p. 126).
- [373] T. Shen, A. Kar, and S. Fidler. "Learning to caption images through a lifetime by asking questions". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 10393–10402 (see p. 5).

- [374] R. Shetty, M. Rohrbach, L. Anne Hendricks, M. Fritz, and B. Schiele. "Speaking the same language: Matching machine to human captions by adversarial training". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 4135–4144 (see p. 99).
- [375] W. Shimoda and K. Yanai. "Distinct class-specific saliency maps for weakly supervised semantic segmentation". In: *European Conference on Computer Vision*. Springer. 2016, pp. 218–234 (see p. 128).
- [376] Z. Shu, M. Sahasrabudhe, R. Alp Guler, D. Samaras, N. Paragios, and I. Kokkinos. "Deforming Autoencoders: Unsupervised Disentangling of Shape and Appearance". In: *The European Conference on Computer Vision (ECCV)*. 2018 (see p. 97).
- [377] K. Shuster, S. Humeau, H. Hu, A. Bordes, and J. Weston. "Engaging image captioning via personality". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12516–12526 (see p. 99).
- [378] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014) (see pp. 18, 22, 32, 33, 41, 42, 107, 135).
- [379] S. N. Sinha, K. Ramnath, and R. Szeliski. "Detecting and Reconstructing 3D Mirror Symmetric Objects". In: *European Conference on Computer Vision (ECCV)*. 2012 (see p. 28).
- [380] F. H. Sinz, J. Q. Candela, G. H. Bakır, C. E. Rasmussen, and M. O. Franz. "Learning depth from stereo". In: *Pattern Recognition*. Springer, 2004, pp. 245–252 (see p. 27).
- [381] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng. "Grounded compositional semantics for finding and describing images with sentences". In: *Transactions of the Association for Computational Linguistics* 2 (2014), pp. 207–218 (see pp. 101, 105).
- [382] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958 (see p. 19).
- [383] R. K. Srivastava, K. Greff, and J. Schmidhuber. "Highway networks". In: *arXiv preprint arXiv:1505.00387* (2015) (see p. 130).
- [384] C.-N. Straehle, U. Koethe, G. Knott, K. Briggman, W. Denk, and F. A. Hamprecht. "Seeded watershed cut uncertainty estimators for guided interactive segmentation". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pp. 765–772 (see p. 128).
- [385] F. Strub, H. De Vries, J. Mary, B. Piot, A. Courville, and O. Pietquin. "End-to-end optimization of goal-driven and visually grounded dialogue systems". In: *arXiv preprint arXiv:1703.05423* (2017) (see p. 131).
- [386] F. Strub, M. Seurin, E. Perez, H. de Vries, J. Mary, P. Preux, and A. CourvilleOlivier Pietquin. "Visual reasoning with multi-hop feature modulation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 784–800 (see p. 130).
- [387] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. "A benchmark for the evaluation of RGB-D SLAM systems". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 573–580 (see p. 65).
- [388] S. Subramanian, S. R. Mudumba, A. Sordoni, A. Trischler, A. C. Courville, and C. Pal. "Towards Text Generation with Adversarially Learned Neural Outlines". In: *Advances in Neural Information Processing Systems*. 2018, pp. 7562–7574 (see p. 110).
- [389] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. "On the importance of initialization and momentum in deep learning". In: *International conference on machine learning*. 2013, pp. 1139–1147 (see p. 17).

- [390] I. Sutskever, O. Vinyals, and Q. V. Le. "Sequence to sequence learning with neural networks". In: *Advances in neural information processing systems*. 2014, pp. 3104–3112 (see pp. 102, 103).
- [391] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. 2018 (see p. 6).
- [392] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. "Policy gradient methods for reinforcement learning with function approximation". In: *Advances in neural information processing systems*. 2000, pp. 1057–1063 (see p. 110).
- [393] S. Suwajanakorn and C. Hernandez. "Depth from Focus with Your Mobile Phone". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015 (see p. 26).
- [394] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9 (see pp. 18, 19).
- [395] R. Sznitman, K. Ali, R. Richa, R. H. Taylor, G. D. Hager, and P. Fua. "Data-driven visual tracking in retinal microsurgery". In: *MICCAI*. Springer, 2012, pp. 568–575 (see pp. 73, 78).
- [396] R. Sznitman, C. Becker, and P. Fua. "Fast Part-Based Classification for Instrument Detection in Minimally Invasive Surgery". In: *MICCAI*. Springer. 2014 (see p. 79).
- [397] R. Sznitman, R. Richa, R. H. Taylor, B. Jedynek, and G. D. Hager. "Unified detection and tracking of instruments during retinal microsurgery". In: *IEEE trans. on Pattern Analysis and Machine Intelligence* 35.5 (2013) (see p. 73).
- [398] M. Tang, F. Perazzi, A. Djelouah, I. Ben Ayed, C. Schroers, and Y. Boykov. "On regularized losses for weakly-supervised cnn segmentation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 507–522 (see p. 128).
- [399] K. Tateno, F. Tombari, I. Laina, and N. Navab. "CNN-SLAM: Real-time dense monocular slam with learned depth prediction". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6243–6252 (see pp. 56, 59, 66).
- [400] K. Tateno, F. Tombari, I. Laina, and N. Navab. "CNN-SLAM: Real-time dense monocular slam with learned depth prediction". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6243–6252 (see p. 155).
- [401] K. Tateno, F. Tombari, and N. Navab. "Real-time and scalable incremental segmentation on dense slam". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 4465–4472 (see p. 65).
- [402] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. "The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pp. 103–110 (see p. 25).
- [403] I. Laina*, N. Rieke*, C. Rupprecht, J. P. Vizcaíno, A. Eslami, F. Tombari, and N. Navab. "Concurrent segmentation and localization for tracking of surgical instruments". In: *International conference on medical image computing and computer-assisted intervention*. Springer. 2017, pp. 664–672 (see p. 155).
- [404] I. Laina*, C. Rupprecht*, V. Belagiannis, F. Tombari, and N. Navab. "Deeper depth prediction with fully convolutional residual networks". In: *2016 Fourth international conference on 3D vision (3DV)*. IEEE. 2016, pp. 239–248 (see p. 155).
- [405] I. Laina, C. Rupprecht, and N. Navab. "Towards Unsupervised Image Captioning with Shared Multimodal Embeddings". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 7414–7424 (see p. 155).
- [406] S. Thorpe, D. Fize, and C. Marlot. "Speed of processing in the human visual system". In: *nature* 381.6582 (1996), pp. 520–522 (see p. 7).

- [407] S. Thrun and B. Wegbreit. "Shape From Symmetry". In: *International Conference on Computer Vision (ICCV)*. 2005 (see p. 28).
- [408] S. Thrun and B. Wegbreit. "Shape from symmetry". In: *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. Vol. 2. IEEE. 2005, pp. 1824–1831 (see p. 26).
- [409] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. "Efficient object localization using convolutional networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 648–656 (see p. 75).
- [410] F. Tosi, F. Aleotti, M. Poggi, and S. Mattocchia. "Learning monocular depth estimation infusing traditional stereo knowledge". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9799–9809 (see p. 31).
- [411] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. "Bundle adjustment: A modern synthesis". In: *International workshop on vision algorithms*. Springer. 1999, pp. 298–372 (see p. 58).
- [412] J. R. Uijlings, M. Andriluka, and V. Ferrari. "Panoptic Image Annotation with a Collaborative Assistant". In: *arXiv preprint arXiv:1906.06798* (2019) (see pp. 126, 129).
- [413] S. Ullman. "The interpretation of structure from motion". In: *Proceedings of the Royal Society of London. Series B. Biological Sciences* 203.1153 (1979), pp. 405–426 (see pp. 26, 28).
- [414] D. Ulyanov, A. Vedaldi, and V. Lempitsky. "Instance normalization: The missing ingredient for fast stylization". In: *arXiv preprint arXiv:1607.08022* (2016) (see p. 19).
- [415] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox. "Demon: Depth and motion network for learning monocular stereo". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 5038–5047 (see p. 59).
- [416] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. Van Gool. "Learning to classify images without labels". In: *arXiv preprint arXiv:2005.12320* (2020) (see p. 97).
- [417] J. Varley, C. DeChant, A. Richardson, A. Nair, J. Ruales, and P. Allen. "Shape completion enabled robotic grasping". In: *arXiv preprint arXiv:1609.08546* (2016) (see p. 81).
- [418] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008 (see p. 99).
- [419] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. "Cider: Consensus-based image description evaluation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 4566–4575 (see pp. 99, 113).
- [420] S. Venugopalan, L. Anne Hendricks, M. Rohrbach, R. Mooney, T. Darrell, and K. Saenko. "Captioning images with diverse objects". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 5753–5761 (see p. 100).
- [421] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. "Sequence to sequence-video to text". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 4534–4542 (see p. 98).
- [422] U. Viereck, A. Pas, K. Saenko, and R. Platt. "Learning a visuomotor controller for real world robotic grasping using simulated depth images". In: *Conference on Robot Learning*. 2017 (see p. 81).
- [423] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. "Show and tell: A neural image caption generator". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3156–3164 (see pp. 98, 108, 112, 113, 123).

- [424] L. Von Ahn, M. Kedia, and M. Blum. “Verbosity: a game for collecting common-sense facts”. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*. 2006, pp. 75–78 (see p. 5).
- [425] H. de Vries, F. Strub, S. Chandar, O. Pietquin, H. Larochelle, and A. Courville. “Guesswhat?! visual object discovery through multi-modal dialogue”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 5503–5512 (see p. 131).
- [426] H. de Vries, F. Strub, J. Mary, H. Larochelle, O. Pietquin, and A. C. Courville. “Modulating early visual processing by language”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 6594–6604 (see pp. 129, 130).
- [427] C. Wang, J. Miguel Buenaposada, R. Zhu, and S. Lucey. “Learning depth from monocular videos using direct methods”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2022–2030 (see pp. 30, 60).
- [428] G. Wang, M. A. Zuluaga, W. Li, R. Pratt, P. A. Patel, M. Aertsen, T. Doel, A. L. David, J. De-prest, S. Ourselin, et al. “Deepigeos: A deep interactive geodesic framework for medical image segmentation”. In: *arXiv preprint arXiv:1707.00652* (2017) (see p. 129).
- [429] G. Wang, W. Li, M. A. Zuluaga, R. Pratt, P. A. Patel, M. Aertsen, T. Doel, A. L. David, J. De-prest, S. Ourselin, et al. “Interactive medical image segmentation using deep learning with image-specific fine tuning”. In: *IEEE transactions on medical imaging* 37.7 (2018), pp. 1562–1573 (see p. 129).
- [430] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan. “Learning to detect salient objects with image-level supervision”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 136–145 (see p. 128).
- [431] L. Wang, A. Schwing, and S. Lazebnik. “Diverse and accurate image description using a variational auto-encoder with an additive gaussian encoding space”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 5756–5766 (see p. 99).
- [432] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. L. Yuille. “Towards unified depth and semantic prediction from a single image”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 2800–2809 (see pp. 29, 30, 45).
- [433] S. Wang, R. Clark, H. Wen, and N. Trigoni. “Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 2043–2050 (see p. 59).
- [434] T. Wang, B. Han, and J. Collomosse. “Touchcut: Fast image and video segmentation using single-touch interaction”. In: *Computer Vision and Image Understanding* 120 (2014), pp. 14–30 (see p. 128).
- [435] X. Wang, K. Yu, C. Dong, and C. Change Loy. “Recovering realistic texture in image super-resolution by deep spatial feature transform”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 606–615 (see p. 130).
- [436] Z. Wang, Z. Li, B. Wang, and H. Liu. “Robot grasp detection using multimodal deep convolutional neural networks”. In: *Advances in Mechanical Engineering* (2016) (see pp. 81, 82, 86, 88).
- [437] J. Watson, M. Firman, G. J. Brostow, and D. Turmukhambetov. “Self-Supervised Monocular Depth Hints”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 2162–2171 (see p. 31).
- [438] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. “Convolutional pose machines”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (see p. 75).

- [439] Y. Wei, X. Liang, Y. Chen, X. Shen, M.-M. Cheng, J. Feng, Y. Zhao, and S. Yan. "Stc: A simple to complex framework for weakly-supervised semantic segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* 39.11 (2016), pp. 2314–2320 (see p. 128).
- [440] J. Weiss, N. Rieke, M. A. Nasser, M. Maier, A. Eslami, and N. Navab. "Fast 5DOF needle tracking in iOCT". In: *International journal of computer assisted radiology and surgery* 13.6 (2018), pp. 787–796 (see p. 73).
- [441] R. J. Williams. "Simple statistical gradient-following algorithms for connectionist reinforcement learning". In: *Machine learning* 8.3-4 (1992), pp. 229–256 (see pp. 99, 110).
- [442] P. Wohlhart and V. Lepetit. "Learning descriptors for object recognition and 3d pose estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 3109–3118 (see p. 25).
- [443] R. J. Woodham. "Photometric method for determining surface orientation from multiple images". In: *Optical engineering* 19.1 (1980), p. 191139 (see p. 26).
- [444] C. Wu. "VisualSFM: A visual structure from motion system". In: (2011) (see p. 28).
- [445] Q. Wu, C. Shen, L. Liu, A. Dick, and A. Van Den Hengel. "What value do explicit high level concepts have in vision to language problems?". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 203–212 (see p. 99).
- [446] S. Wu, C. Rupprecht, and A. Vedaldi. "Unsupervised learning of probably symmetric deformable 3d objects from images in the wild". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 1–10 (see p. 28).
- [447] S. Wu, J. Wieland, O. Farivar, and J. Schiller. "Automatic alt-text: Computer-generated image descriptions for blind users on a social network service". In: *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. ACM. 2017, pp. 1180–1192 (see p. 96).
- [448] Y. Wu and K. He. "Group normalization". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 3–19 (see p. 19).
- [449] K. Xian, C. Shen, Z. Cao, H. Lu, Y. Xiao, R. Li, and Z. Luo. "Monocular relative depth perception with web stereo data supervision". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 311–320 (see p. 29).
- [450] J. Xiao, J. Hays, B. C. Russell, G. Patterson, K. Ehinger, A. Torralba, and A. Oliva. "Basic level scene understanding: categories, attributes and structures". In: *Frontiers in psychology* 4 (2013), p. 506 (see p. 7).
- [451] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun. "Unified perceptual parsing for scene understanding". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 418–434 (see p. 7).
- [452] J. Xie, R. Girshick, and A. Farhadi. "Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks". In: *European Conference on Computer Vision*. Springer. 2016, pp. 842–857 (see p. 30).
- [453] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. "Aggregated residual transformations for deep neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1492–1500 (see p. 19).
- [454] C. Xu and J. L. Prince. "Snakes, shapes, and gradient vector flow". In: *IEEE Transactions on image processing* 7.3 (1998), pp. 359–369 (see p. 128).

- [455] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe. “Multi-scale continuous crfs as sequential deep networks for monocular depth estimation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 5354–5362 (see p. 29).
- [456] D. Xu, W. Ouyang, X. Wang, and N. Sebe. “Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 675–684 (see p. 30).
- [457] D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci. “Structured attention guided convolutional neural fields for monocular depth estimation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3917–3925 (see p. 29).
- [458] J. Xu, A. G. Schwing, and R. Urtasun. “Learning to segment under various forms of weak supervision”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3781–3790 (see p. 128).
- [459] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. “Show, attend and tell: Neural image caption generation with visual attention”. In: *International Conference on Machine Learning*. 2015, pp. 2048–2057 (see pp. 98, 133).
- [460] N. Xu, B. Price, S. Cohen, J. Yang, and T. S. Huang. “Deep interactive object selection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 373–381 (see p. 128).
- [461] G. Yang, P. Hu, and D. Ramanan. “Inferring distributions over depth from a single image”. In: *arXiv preprint arXiv:1912.06268* (2019) (see p. 29).
- [462] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh. “Visual Curiosity: Learning to Ask Questions to Learn Visual Recognition”. In: *2nd Annual Conference on Robot Learning, CoRL 2018, Zürich, Switzerland, 29-31 October 2018, Proceedings*. 2018, pp. 63–80 (see pp. 5, 148).
- [463] L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos. “Efficient video object segmentation via network modulation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 6499–6507 (see p. 130).
- [464] N. Yang, R. Wang, J. Stuckler, and D. Cremers. “Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 817–833 (see pp. 31, 59).
- [465] Z. Yang, P. Wang, W. Xu, L. Zhao, and R. Nevatia. “Unsupervised Learning of Geometry From Videos With Edge-Aware Depth-Normal Consistency”. In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. 2018, pp. 7493–7500 (see p. 30).
- [466] Z. Yang, Y. Yuan, Y. Wu, W. W. Cohen, and R. R. Salakhudinov. “Review networks for caption generation”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 2361–2369 (see p. 99).
- [467] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. “Describing videos by exploiting temporal structure”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 4507–4515 (see p. 98).
- [468] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei. “Boosting image captioning with attributes”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 4894–4902 (see p. 99).
- [469] T. Yao, Y. Pan, Y. Li, and T. Mei. “Exploring visual relationship for image captioning”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 684–699 (see p. 99).

- [470] T. Yao, Y. Pan, Y. Li, and T. Mei. "Incorporating copying mechanism in image captioning for learning novel objects". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6580–6588 (see p. 99).
- [471] Z. Yin and J. Shi. "Geonet: Unsupervised learning of dense depth, optical flow and camera pose". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1983–1992 (see p. 30).
- [472] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. "Image captioning with semantic attention". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 4651–4659 (see p. 99).
- [473] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg. "Modeling context in referring expressions". In: *European Conference on Computer Vision*. Springer. 2016, pp. 69–85 (see p. 131).
- [474] B. S. Zapata-Impata. "Using Geometry to Detect Grasping Points on 3D Unknown Point Cloud". In: *International Conference on Informatics in Control, Automation and Robotics*. 2017 (see p. 81).
- [475] M. D. Zeiler and R. Fergus. "Visualizing and understanding convolutional networks". In: *European conference on computer vision*. Springer. 2014, pp. 818–833 (see pp. 32, 34).
- [476] M. D. Zeiler, G. W. Taylor, and R. Fergus. "Adaptive deconvolutional networks for mid and high level feature learning". In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE. 2011, pp. 2018–2025 (see p. 32).
- [477] H. Zhan, R. Garg, C. Saroj Weerasekera, K. Li, H. Agarwal, and I. Reid. "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 340–349 (see pp. 30, 59).
- [478] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas. "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 5907–5915 (see p. 126).
- [479] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros. "Real-time user-guided image colorization with learned deep priors". In: *arXiv preprint arXiv:1705.02999* (2017) (see p. 129).
- [480] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah. "Shape-from-shading: a survey". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 21.8 (1999), pp. 690–706 (see p. 26, 28).
- [481] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. "Pyramid scene parsing network". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2881–2890 (see p. 128).
- [482] S. Zhao, H. Fu, M. Gong, and D. Tao. "Geometry-Aware Symmetric Domain Adaptation for Monocular Depth Estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9788–9798 (see p. 31).
- [483] W. Zhao, W. Xu, M. Yang, J. Ye, Z. Zhao, Y. Feng, and Y. Qiao. "Dual learning for cross-domain image captioning". In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM. 2017, pp. 29–38 (see p. 100).
- [484] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus. "Simple baseline for visual question answering". In: *arXiv preprint arXiv:1512.02167* (2015) (see p. 131).
- [485] H. Zhou, B. Ummenhofer, and T. Brox. "Deeptam: Deep tracking and mapping". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 822–838 (see p. 59).
- [486] J. Zhou and S. Payandeh. "Visual tracking of laparoscopic instruments". In: *J. Autom. Cont. Eng. Vol. 2.3* (2014), pp. 234–241 (see p. 73).

- [487] M. Zhou, X. Wang, J. Weiss, A. Eslami, K. Huang, M. Maier, C. P. Lohmann, N. Navab, A. Knoll, and M. A. Nasser. "Needle Localization for Robot-assisted Subretinal Injection based on Deep Learning". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 8727–8732 (see p. 73).
- [488] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. "Unsupervised learning of depth and ego-motion from video". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 1851–1858 (see pp. 30, 59).
- [489] X. Zhou, X. Lan, H. Zhang, Z. Tian, Y. Zhang, and N. Zheng. "Fully convolutional grasp detection network with oriented anchor box". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 7223–7230 (see p. 89).
- [490] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2223–2232 (see p. 97).
- [491] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 19–27 (see p. 101).
- [492] Y. Zou, Z. Luo, and J.-B. Huang. "Df-net: Unsupervised joint learning of depth and flow using cross-task consistency". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 36–53 (see p. 30).
- [493] L. Zwald and S. Lambert-Lacroix. "The BerHu penalty and the grouped effect". In: *arXiv preprint arXiv:1207.6868* (2012) (see p. 37).

List of Figures

3.1	Neural network architectures. a) An MLP as a fully-connected feed-forward neural network. b) A CNN with 3×1 convolutions with zero padding and stride 1 and 2 respectively. A neuron in the second layer has a receptive field of 5 and thus “sees” the entire input vector. The last layer is fully-connected. c) A simple RNN; the hidden state h_t depends on the state at time step $t - 1$	15
3.2	Equations and schematic of an LSTM.	20
3.3	Equations and schematic of a GRU.	21
4.1	FCRN architecture. The architecture extends ResNet-50 to a fully convolutional model. The fully-connected layer is replaced by the proposed up-sampling layers, which result in an output of (approximately) half of the input resolution.	33
4.2	Comparison of upsampling blocks. (a) Standard up-convolution. (b) Faster up-convolution. (c) The proposed up-projection. (d) Faster up-projection. . .	35
4.3	Faster up-convolution. Top part: 2×2 unpooling, followed by a 5×5 convolution, resulting in a feature map which is doubled in width and height. In a sliding window fashion, depending on the position of the filter, there are four unique constellations (A,B,C,D), ignoring zero values. Bottom part: one could convolve the low-resolution feature map with the four distinct filters—their shape is decided by the non-zero pixel groups of the high-resolution feature map. It is then possible to interleave the elements of the resulting feature maps to reach an equivalent outcome as the top part, while avoiding zero multiplications. Note: A,B,C,D in the kernels only indicate pixel groups; the actual weight values are not uniform.	36
4.4	Depth distribution in common benchmarks (a) NYU Depth v2 and (b) Make3D. We plot the histograms of discretized depth values. The x-axis indicates depth in meters. In (b) the two spikes around 80m correspond to sky pixels.	37
4.5	The Berhu function \mathcal{L}_B for $\alpha = 1$ in comparison to \mathcal{L}_2 and \mathcal{L}_1	37
4.6	Depth predictions on NYU Depth v2. We compare the qualitative results among baselines AlexNet (UpConv), VGG-16 (UpConv), ResNet-50 (FC), and the proposed FCRN. We further compare to the publicly available predictions of Eigen and Fergus [93]. In the colormap, blue is near, while red is far. For better comparisons, all predictions are shown with respect to the ground truth colormap, thus respecting the real scale.	44
4.7	Depth Prediction on Make3D. Displayed are RGB images (first row), ground truth depth maps (middle row) and our predictions (last row). Pixels that correspond to distances $> 70\text{m}$ in the ground truth are masked out	46

4.8	3D scene reconstructions from an RGB-(learned) D SLAM framework. In contrast to standard SLAM methods that require an RGB-D camera, we use monocular sequences and predict the corresponding depth map for each frame using our method. The reconstructions are obtained for a kitchen sequence of NYU Depth v2. We compare the reconstructions based on predictions from AlexNet, VGG, FCRN as well as using the Kinect-v2 depth measurements provided by the dataset (ground truth).	48
4.9	Application to synthetic defocus. We predict a depth map from a single image and use it to render a synthetically defocused image at various depths. The result is achieved by applying a bilateral filter based on the distance to the plane set in focus by the user.	49
4.10	Semantic segmentation on NYU Depth v2. We show 4-class and 40-class predictions and the corresponding ground truth labels.	51
5.1	Overview of CNN-SLAM. A CNN predicts depth and semantics for keyframes, which are then fused into a <i>geometric</i> SLAM framework for camera pose estimation and frame-wise refinement.	61
5.2	Component evaluation. Comparison of reconstruction performance using (a) raw depth predictions, (b) focal length adjustment (Equation (5.5)) and (c) adjustment and frame-wise depth refinement (Section 5.3.3), in terms of (A) pose trajectory accuracy and (B) depth estimation accuracy. In the binary maps (B), blue indicates correctly estimated depth (within 10 % of the ground truth depth values) and red indicates wrong predictions. The comparison is shown on a sample sequence of the ICL-NUIM dataset [147].	64
5.3	Comparing depth map density and accuracy on the <i>office2</i> sequence of ICL-NUIM. FCRN shows the raw depth prediction for the selected frame, while CNN-SLAM is the result after keyframe refinement. The reported % are the correctly estimated depth pixels on this frame.	66
5.4	Geometric and semantic reconstructions. We show the geometric and semantic reconstructions of our method on three scenes: NYU-Depth-v2 <i>bedroom_0115</i> (left), <i>kitchen_0046</i> (middle) and an office scene from a custom setup (right). Camera trajectory is shown in green.	67
5.5	Reconstruction under pure rotational motion. We compare LSD-SLAM and CNN-SLAM reconstructions of a sequence (TUM-RGBD <i>fr1/rpy</i>) that consists of rotational motion.	67
5.6	Reconstruction scale. We augment an object (skeleton) in the reconstructed environments of LSD-SLAM and CNN-SLAM to show the important of real-scale reconstructions in applications such as augmented reality.	68
6.1	Overview of the proposed method. We jointly predict a segmentation map and the landmark positions of a surgical tool by adapting FCRN.	73

6.2	Concurrent segmentation and localization (CSL) architecture. We also show two baselines: the single-task localization-only model (L) and the multi-task localization-and-segmentation model (SL) where only encoder weights are shared among the two tasks. Instead, our proposed architecture CSL treats the two tasks with the same output dimensionality and all weights in the network (except for the prediction layers) are shared.	74
6.3	Ablation experiments on the Retinal Microsurgery dataset. We plot and compare the localization accuracy of the baseline models (L, SL), U-net [347] and the proposed method (CSL) for (a) the left tip, (b) the right tip and (c) the center joint of the instrument.	78
6.4	Comparisons on RM dataset. We compare our method to FPBC [396], POSE [344] and Online Adaption [345]. <i>Half Split:</i> (a) to (c) correspond to the <i>KBB</i> scores of left tip, right tip and center joint. <i>Cross Validation:</i> (d) shows the average <i>KBB</i> score for the center point over all folds.	79
6.5	Qualitative results on EndoVis 2015. The semantic segmentation (manipulator/shaft) and joint localization are overlaid on the original image. <i>Left:</i> We show a representative frame from a test sequence containing one tool. <i>Right:</i> Test sequence contains two tools, although only the right tool exists in the training data. Due to proper augmentation during training, the model is able to generalize well.	80
6.6	Reformulating oriented rectangles as heatmaps. The heatmaps are created as Gaussian mixtures with the plate centers as means and the variance σ_x proportional to the gripper height (σ_y is a chosen constant).	83
6.7	Multiple grasp configurations for an object. We show all annotated oriented rectangles for an image of the Cornell grasp detection dataset [235] and the corresponding heatmaps (our representation).	83
6.8	The Cornell Grasp dataset [235]. We show some sample (center-cropped) images from the dataset.	87
6.9	Predicted heatmaps from our model with $M = 5$. We also compare to the single-prediction model (right). A solid frame around heatmaps indicates a successful grasp, while a dashed line indicates a missed detection. (\checkmark) indicates the top-ranked hypothesis. The rectangles resulting from top-ranked heatmaps are overlaid on the RGB image (magenta) against the closest ground truth (green).	89
6.10	Generalization to common household objects. We show the top-ranked heatmap predicted for several self-recorded images. Objects 1-5 are similar in shape to the objects in the Cornell dataset, but objects 6-12 show novel shapes and textures.	90
6.11	Grasp rectangle prediction. We show the diversity of the predicted grasp configurations, after they have been converted from heatmaps to oriented rectangles.	91

7.1	Method overview. We learn a joint embedding space of text and image features from <i>disjoint</i> language and image domains. The embedding space is structured by visual concepts and their co-occurrence. In the language domain, visual words are <u>underlined</u> . A shared decoder can decode image and text embeddings in the same manner and thus unsupervised captioning can be achieved when the two domains are aligned.	102
7.2	Bags of visual words. We show two sample sentences and simplified part-of-speech tagging. We extract synsets from the detected visual nouns to create bags $\mathcal{W}_1, \mathcal{W}_2$ and use those to denote the two sentences as <i>positive</i> to each other since they have two synsets in common (<code>man.n.01</code> and <code>bicycle.n.01</code>). . . .	104
7.3	Language model. We build an encoder-decoder language model which, in addition to sentence reconstruction, is trained with sentence triplets. The triplet loss creates an embedding space that is structured by the visual concepts that are present in the language domain.	105
7.4	Domain alignment model. For each image we sample a set of candidate sentences, which are encoded using the previously trained language model. The image features are projected into the same embedding space, minimizing a robust loss towards the sentence embeddings, a conditional adversarial loss and the sentence reconstruction loss.	107
7.5	Example on image-caption correspondences. The image is overlaid with detected object categories from MSCOCO. The candidate sentences are real annotations from MSCOCO. Although all of them have at least two entities in common with the image, they exhibit very different degrees of correlation. . . .	109
7.6	Image captioning with the alignment-only model. We show predicted captions for MSCOCO test images. For this model we only train the translation from image features to the shared embedding space but do not finetune the decoder. The proposed objectives (\mathcal{L}_R and \mathcal{L}_{adv}) perform better at this mapping than the \mathcal{L}_2 baseline.	117
7.7	Image captioning with the joint model. We show predicted captions for MSCOCO test images. COCO and OID are results from our <i>unpaired</i> model, while GCC and VQA refer to the <i>unsupervised</i> model trained on MSCOCO images using as sentence corpus the Conceptual Captions and VQA-v2 datasets respectively.	118
7.8	Additional results on unsupervised captioning. We show more examples from three different sentence corpora.	120
7.9	Visualization of the language model's embedding space. We compare the t-SNE embeddings of the latent space created by our language model with a) visually structured training with the triplet loss and b) standard maximum likelihood training.	121
7.10	Visualization of the shared embedding space. We show the t-SNE projection of the multimodal embeddings. Zooming into one cluster we show that sentences embeddings [L] lie in visual-semantic groups with image embeddings [I].	122

8.1	Overview of user-network interaction for semantic image segmentation. We introduce a module capable of improving the predictions of a CNN by guiding its activations conditioned on user-provided hints. The means of communication is natural language which is processed by an RNN. In this example, the base model mistakenly labels the top of the image as <i>sky</i> instead of <i>clouds</i> . When the user indicates this mistake, inference is revisited and the prediction is updated accordingly without any change of the network’s parameters. . . .	127
8.2	Guiding a network with natural language hints. The word embeddings of the hint sequence are encoded by an RNN, producing the guiding vectors α , β , γ (biases omitted from the visual illustration for simplicity). The guiding vectors modify the CNN features of a chosen layer, thus forcing a change in the final prediction, without any other parameter update. While a real user can interact with the network at test time, during training we simulate the user by programmatically generated hints.	137
8.3	Query Generator. We illustrate the process to automatically generate queries to substitute the user during training.	138
8.4	Qualitative results using <i>find</i> hints. The guided network is able to recover from ambiguities and difficult cases, such as heavily occluded objects, and also refine partially predicted segments.	143
8.5	Qualitative results using <i>remove</i> hints. While the user specifies which category to remove, they do not specify what to replace it with. In most cases this is inferred by context.	144
8.6	Qualitative example of repeated guiding. We provide multiple hints to the network to keep correcting mistakes with respect to confused classes <i>sky/clouds</i> , <i>playing field/ground</i>	146
8.7	Visualization of γ-vectors. We show the t-SNE plot of the learned semantic vectors for all categories in COCO-Stuff. In this plot the same <i>find</i> hint is encoded for each category. The colormap corresponds to super-categories. . .	147
8.8	Visualization of all guiding vectors. We tile the guiding vectors to visualize them as a 2D heatmap. Two examples are shown: (a) Given the hint, the refined prediction is correct. (b) The missing category is not recovered, even after the hint, although the heatmap suggests that the guide has the right focus. . . .	147

List of Tables

4.1	Comparison of different architectures on NYU Depth v2. We compare different encoders (AlexNet [215], VGG-16 [378], ResNet-50 [151]), up-sampling strategies (FC, UpConv, UpProj) and loss functions ($\mathcal{L}_2, \mathcal{L}_B$). For the reported errors rel, rmse, \log_{10} lower is better, whereas for the accuracies $\delta_j < 1.25^j$ higher is better.	42
4.2	Comparison to prior work on the NYU Depth v2 dataset. For the error metrics (rel, rmse) lower is better, while for the accuracies δ_j higher is better.	45
4.3	Comparison with prior work on Make3D. We report the performance of our models trained with \mathcal{L}_2 and \mathcal{L}_B losses.	46
4.4	4-class semantic indoor segmentation (labeled NYU Depth v2). We compare to prior work that use, additionally to the color image (RGB), depth (D) and surface normals (N) as input to the network.	50
4.5	40-class semantic indoor segmentation (labeled NYU Depth v2). The performance of our model is comparable to concurrent work [298]. 40-class is a much more challenging setting, which includes rare of small object categories; we show that the size of the input image makes a significant difference with respect to all metrics.	50
5.1	Comparison with state-of-the-art SLAM methods. We compare our full approach (CNN-SLAM) and baseline (FCRN) with previous methods on two datasets (ICL-NUIM and TUM-RGBD) under two metrics: absolute trajectory error (in meters) and the percentage of accurately estimated depth pixels. (TUM-RGBD seq1: <i>fr3/long_office_household</i> , seq2: <i>fr3/nostructure_texture_near_withloop</i> and seq3: <i>fr3/structure_texture_far</i>	66
6.1	Segmentation accuracy on the Retinal Microsurgery dataset. We compare the DICE scores of U-net [347] and baselines SL and CSL without long-range skip connections to the proposed model (CSL).	79
6.2	Quantitative comparisons on EndoVis 2015. We evaluate segmentation using recall (Rec), specificity (Spec), balanced accuracy (B.Acc) and DICE scores. Binary evaluation is done after merging the two semantic classes (shaft and grasper). We also report the localization error for the two tools as the Euclidean distance between the predicted and the ground truth center joint of each tool.	80
6.3	Comparison with the state of the art on grasp detection accuracy. <i>Single</i> predicts a single heatmap, while <i>multiple</i> refers to our MHP models. The top-ranked hypothesis is evaluated for the MHP models.	88

6.4	Average grasp estimation accuracy (%) across hypotheses. For the lower limit we evaluate <i>all</i> hypotheses; for the upper limit any correct hypothesis (oracle) leads to grasp success.	91
7.1	Ablation experiments on MSCOCO test set [188]. We do not use pairs of images and captions when training. MSCOCO ground truth object categories are used as visual concepts in the image domain. All proposed components positively add to the performance on the captioning task.	114
7.2	Comparison with the state of the art on COCO test set [188]. We evaluate under the <i>unpaired</i> setting of [106], using OID [221] object categories for visual concept extraction. We use beam search of size 3 and in the last two rows we further constrain it on target words (object categories) for an additional improvement.	116
7.3	Evaluation under the unsupervised setting. Image and captions come from independent sources.	119
8.1	Performance after a number of pixel queries. We apply guiding by back-propagation on a pre-trained FCN-8s [260] on the PascalVOC 2012 <i>val</i> set [99]. We evaluate the model in terms of mean intersection over union (mIoU) and pixel accuracy after a number of interactions, observing constant improvement.	136
8.2	Guiding module variations. We report the mean intersection over union (mIoU) metric for different implementations of the guiding module. In all experiments we guide layer <code>res4a</code> using <code>find</code> hints.	141
8.3	Guiding location. We evaluate the effect of different guiding locations (layers) in terms of mIoU performance. In all experiments <code>find</code> hints are used.	141
8.4	Hint functionality. We compare the performance gain when training guides with different types of hints and their combination.	142
8.5	Semantic categories of COCO-Stuff. We visualize all semantic categories in the dataset as well as the colormap used in the remaining figures.	145
8.6	Repeated guiding. We guide the network repeatedly with several <code>find</code> or <code>rmv</code> hints, each one depending on the latest prediction. After three hints, the performance gain decreases because the guide, which has been only trained with a single hint, over-modulates the features.	146

