Maschinelles Sehen für die automatische Erkennung von Baubehelfen

Katrin Jahr und Alex Braun

Lehrstuhl für Computergestützte Modellierung und Simulation,
Technische Universität München, Arcisstr. 21, 80333 München
E-Mail: katrin.jahr@tum.de, alex.braun@tum.de

Baustellenmonitoring als entscheidender Bestandteil der Bauausführung wird zurzeit vorrangig händisch ausgeführt. Moderne Methoden der Bildverarbeitung bieten einen vielversprechenden Ansatz zur Reduzierung der manuellen Arbeit vor Ort. Während maschinelle Lernalgorithmen, wie Convolutional Neuronal Networks, in anderen Anwendungsbereichen weit verbreitet sind, werden sie bisher von der CAE-Industrie weitgehend vernachlässigt. In diesem Beitrag wird maschinelles Lernen zur Erkennung von Baubehelfen auf Fotografien von Baustellen verwendet. In einer Fallstudie mit 750 Fotos, welche etwa 10.000 Schalungselementen enthalten, konnten Genauigkeiten von 90% bei der Klassifizierung und 60% bei der Lokalisierung von Schalungen erreicht werden.

Keywords: Bildverarbeitung, BIM, UAV, Computer Vision

1 Einleitung

Die fortschreitende Digitalisierung der Bauindustrie bietet vielfältige Möglichkeiten für den Entwurf, die Planung und die Überwachung von Gebäuden. Dabei konzentrierten sich viele Forschungsprojekte auf Methoden des computergestützten Ingenieurbaus und den Planungsprozess. Nach dem Abschluss der Planungsphase wird digitale Unterstützung kaum in Anspruch genommen. Besonders die arbeitsintensive Baufortschrittsüberwachung wird in der Regel mit wenig technischer Unterstützung händisch vor Ort durchgeführt.

Die Bildverarbeitung bietet großes Potential für die Bauüberwachung. In den letzten Jahren wurden verschiedene Bilderfassungsgeräte, wie Drohnen und Laserscanner, auf ihre Eignung in der Bauindustrie untersucht. Mit den resultierenden 3D-Punktwolken und Informationen aus dem zugehörigen BIM-Model kann ein detaillierten Vergleich von Planung („as-planned“) und Ausführung („as-built“) erstellt und der aktuelle Baufortschritt verfolgt werden (Golparvar-fard, Pena-Mora, and Savarese 2009; Braun et al. 2015).

Zur Erzeugung hochwertiger Punktwolken wird eine große Anzahl überlappender Fotografien benötigt. Dies bedeutet einen großen Aufwand sowohl bei der Bilderfassung, als auch bei der Bildverarbeitung. Viele Überwachungsaufgaben, beispielsweise die Überwachung der Baustelleneinrichtung und der Materialvorhaltung, erfordern jedoch keine detaillierten 3D-Informationen. Für diese Anwendungsfälle bietet die Analyse und Objekterkennung einzelner Fotografien eine günstige Alternative, da die Aufnahme einzelner Bilder einen wesentlich geringeren Aufwand darstellt als die Erstellung einer 3D-Punktwolke.

In diesem Paper wird am Beispiel von Schalungselementen ein Ansatz der künstlichen Intelligenz demonstriert, mit dem Bauelemente und Bauhilfsmittel vor Ort erkannt und lokalisiert werden können. Dazu werden zwei verschiedene neuronale Netze verwendet, mit deren Hilfe Fotografien von Baustellen analysiert werden. Der erste Teil des Beitrages gibt einen Überblick über den aktuellen Stand im Bereich der Bildanalyse, gefolgt von einer Beschreibung der verwendeten Methoden. Anschließend wird eine Fallstudie präsentieren und der Beitrag einer Zusammenfassung unserer Ergebnisse abgeschlossen.

2 Stand der Technik

Das maschinelle Sehen erhielt durch die jüngsten Fortschritte im Bereich des autonomen Fahrens und maschinellen Lernens verstärkte Aufmerksamkeit. Die Anwendung von Techniken der Bildanalyse auf Fotografien von Baustellen ist dagegen ein eher neues Thema. Da einer der Schlüsselaspekte des maschinellen Lernens die Sammlung großer Datensätze ist, konzentrieren sich aktuelle Ansätze vorrangig auf die Datenerfassung. Han und Golparvar-Fard (2017) nutzen Amazon Turk zur händischen Generierung von Labels für Bilder. Kropp, Koch, und König (2018) versuchen, am Beispiel von Heizkörpern, Innenausbauteile über deren Ähnlichkeiten zu erkennen.

Für die effektive und effiziente Bildanalyse und Objekterkennung wurden in den letzten Jahrzehnten vermehrt maschinelle Lernalgorithmen eingesetzt. 2012 erreichte das Convolutional Neural Network (CNN) „AlexNet“ (Krizhevsky, Sutskever, and Hinton 2017) einen Top-5-Fehler von 15,3% im renommierten Wettbewerb von ImageNet („Large Scale Visual Recognition Challenge“) (Russakovsky et al. 2015). Diese Fehlerrate galt zum damaligen Zeitpunkt als überraschend gering und zeigt die Überlegenheit von CNN gegenüber anderen Methoden (LeCun, Bengio, and Hinton 2015).

Es gibt verschiedene Aufgaben, die durch Bildverarbeitungsalgorithmen gelöst werden können. Zu den bekannten Problemen gehören die Klassifizierung, bei der Bilder eines Objektes einer bestimmten Klasse zugeordnet werden, die Objekterkennung, bei der mehrere Objekte innerhalb eines Bildes klassifiziert und lokalisiert werden, und die Bildsegmentierung, bei der jedes Pixel eines Bildes klassifiziert wird (Buduma 2017). In diesem Beitrag konzentrieren wir uns auf die Klassifizierung und die Objekterkennung.

CNNs sind in Schichten strukturiert. Jede Schicht umfasst mehrere Berechnungseinheiten (Neuronen), welche mit benachbarten Neuronen in den vorangehenden und nachfolgen Schichten verbunden sind. Die Verbindungen werden während eines Trainingsvorganges gewichtet, wobei die Gewichte innerhalb eines Layers geteilt werden. Die Neuronen der ersten Schicht (input layer) repräsentieren die Pixel des analysierten Bildes, die letzte Schicht (output layer) repräsentiert die möglichen Vorhersagen des Netzes. Zwischen Eingabe- und Ausgabeschicht wird eine beliebige Anzahl von verborgenen Ebenen angeordnet (Abbildung 1).

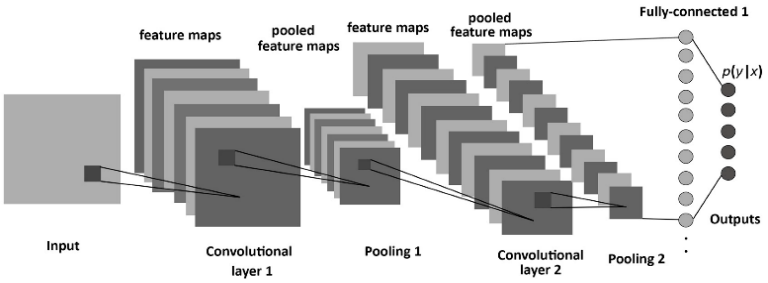

Abbildung 1: Beispiel für ein CNN mit convulutional, pooling und fully conneted layers.

Während AlexNet 8 versteckte Schichten enthält, verwenden GoogLeNet (Szegedy et al. 2015) 22 und Microsoft ResNet (He et al. 2016) mehr als 100 versteckte Schichten. Die Schichten sind in der Regel convolutional layer (Faltung; einzelne Merkmale werden geschärft), pooling layer (überflüssige Informationen werden verworfen) oder fully connected layers (vollständig verbundene Schichten; ermöglichen Klassifizierung) (Buduma 2017; Albelwi and Mahmood 2017).

CNNs können durch Training an verschiedene Probleme, beispielsweise die Erkennung von Schalungselementen, angepasst werden. Während des Trainings wird das Netzwerk mit Bildern gefüttert und die Ergebnisse mit den korrekten Lösungen („ground truth“) verglichen. Anschließend wird die Verbindungstärke zwischen bestimmten Neuronen aufeinanderfolgender Schichten erhöht oder reduziert. Üblicherweise wird das Training über einen Backpropagation Algorithmus durchgeführt (Buduma 2017).

Um ein CNN zu trainieren, werden viele Bilder benötigt, die in einem Preprocessing-Schritt vorbereitet werden müssen. Um das Training beschleunigen, können Gewichte von zuvor trainierten CNNs wiederverwendet werden. Diese können auf einer großen Datenbasis vortrainiert werden – ImageNet stellt etwa 1.000 Bilder pro Klasse zur Verfügung (Russakovsky et al. 2015). Zur Anpassung auf das neue Problem müssen die letzten Schichten des CNN ersetzt werden, und anschließend die Gewichte mit den neuen Daten trainiert werden.

3 Methodik

Dieser Beitrag fokussiert auf die bildbasierte Erkennung von temporären Konstruktionselementen, insbesondere Schalungen. Die Erkennung wiederkehrender, ähnlicher Objekte kann durch maschinelles Lernen gelöst werden. Dabei werden hier die Bildklassifizierung und die Objekterkennung untersucht.

3.1 Bildklassifizierung mit CNNs

Bei der Klassifizierung, die auch als Bilderkennung bezeichnet wird, werden Bilder klassifiziert, die genau ein Objekt enthalten. Jede Klasse, die das CNN erkennen kann, wird durch ein Ausgabe-Neuron repräsentiert. Die Aktivität der Ausgabe-Neuronen wird als die Wahrscheinlichkeit gelesen, dass das Bild ein Objekt der entsprechenden Klasse enthält. Klassifizierungsalgorithmen können nicht zur Analyse von Bildern mit mehreren Objekten verwendet werden. Da Bilder von Baustellen mehr als ein Objekt enthalten, können Bildklassifikationsalgorithmen nur nach der Vorverarbeitung der Daten angewendet werden. Sie können jedoch sehr nützlich sein, um gezielte Fragestellungen zu beantworten, z.B. ob eine Wand an einer bestimmten Position fehlt, derzeit eingeschalt oder bereits fertig gestellt ist.

3.2 Objekterkennung mit CNNs

Die naheliegende Lösung zum Analysieren von Bildern mit mehreren Objekten besteht darin, eine sliding Window Funktion auf das Bild anzuwenden und auf jedem Fenster eine Bildklassifizierung durchzuführen. Diese Methode ist rechnerisch sehr aufwändig. Um den Rechenaufwand zu reduzieren, wurden verschiedene Vorschläge gemacht, z.B. region-proposal networks (z. B. R-CNN (Girshick 2015),(Ren et al. 2017)), die interessante Bereiche innerhalb eines Bildes erkennen und nur diese weiter analysieren, sowie single shot detectors, (zB DetectNet (Tao, Barker, and Sarathy 2016) und YOLO (Redmon et al. 2016; Redmon and Farhadi 2017; Redmon and Farhadi 2018)), die das Bild mit einem Gitter überlagern und jede Zelle analysieren.

3.3 Evaluation von CNNs

Um die Leistung eines Klassifizierungsalgorithmus zu messen, werden der Top-1-Error und der Top-5-Error verwendet. Der Top-1- Error repräsentiert den Anteil der Bilder, für den die korrekte Klasse mit der höchsten Wahrscheinlichkeit vorhergesagt wurde. Der Top-5- Error

ist der Anteil von Bildern, für den die korrekte Klasse innerhalb der 5 Klassen liegt, die mit der höchsten Wahrscheinlichkeit vorhergesagt wurden.

Um die Leistung eines Objektserkennungsalgorithmus zu messen, können die Genauigkeit (precision) p, der Ausfall (recall) r und die durchschnittliche mittlere Genauigkeit (mean average precision) mAP verwendet werden. r und p werden unter Verwendung der Anzahl von richtig positiven Treffern TP, falschen positiven Treffern FP und falsch negativen Treffern FN berechnet: $p = \frac{TP}{TP+FP}$, $r = \frac{TP}{TP+FN}$

Bei Objekterkennungsaufgaben wird eine Vorhersage als richtig positiv gewertet, wenn sich ground truth und Vorhersage weit überlappen. Als Maß wird die „intersection of union (IoU) verwendet, das Verhältnis aus Schnittmenge und Vereinigung der Begrenzungsrechtecke von ground truth und Vorhersage (vgl. Abbildung 2). Üblicherweise zählt eine Vorhersage als richtig positiv, wenn der IoU mehr als 0,5 beträgt. Zusätzlich wird der mAP herangezogen, der über alle Klassen gemittelte Durchschnitt der Genauigkeit in Abhängigkeit des Ausfallwerts (Russakovsky et al. 2015).

$$IoU = \frac{Schnittmenge}{Vereinigung}$$

$$AP = \frac{1}{11} \sum_{r \in \{0.0,\ldots,1.0\}} p_i(r)$$

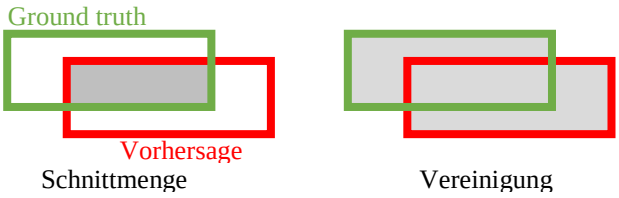

Abbildung 2: Schnittmenge und Vereinigungsmenge für vorhergesagte und tatsächliche Begrenzungsrechtecke

3.4 Labeling

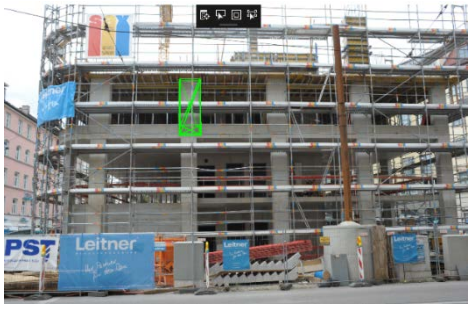

Abbildung 3: Reprojiziertes Begrenzungsrechteck einer Stütze

Das Erstellen der ground truth wird als Labeling bezeichnet. Dabei werden alle Objekte in einem Set von Bildern gekennzeichnet. Da viele Bilder benötigt werden, stellt das händische Labeln einen großen zeitlichen Aufwand dar. Braun et al. 2018 schlagen einen neuen Ansatz zur automatisierten Markierung vor. Im Rahmen des Forschungsprojekts ProgressTrack mit dem Fokus auf automatisierter Fortschrittsüberwachung mit photogrammetrischen Punktwolken wurde ein Algorithmus zur Validierung von Erkennungsergebnissen des Ist-Vergleichs im Vergleich zum geplanten Vergleich entwickelt. Wie in Abbildung 3 dargestellt, kann die projizierte 2D-Geometrie von Konstruktionselementen aus dem Gebäudedatenmodell-Koordinatensystem in das 2D-Koordinatensystem des Bildes umgewandelt werden, in dem das Element enthalten ist. Dies ist möglich, da die Bilder während des photogrammetrischen Prozesses ausgerichtet und orientiert wurden und somit die genaue Position jedes Bildes in Bezug auf das BIM Model bekannt ist. Von dieser Arbeit kann der Prozess des Labelns profitieren, da alle Gebäudeelemente auf allen vorhandenen Bildern markiert werden können. Die Forschung hierzu dauert an.

4 Fallstudie

In den folgenden Abschnitten wird eine Routine zur Bildanalyse mit Datenaufbereitung und Training von CNNs vorgestellt, um Schalungselemente erkennen zu können.

4.1 Vorbereitung der Daten

Als erster Datensatz wurden 9.956 Schalungselemente auf Bildern von drei Baustellen in Deutschland manuell gelabelt. Die Bilder enthalten Schalungselemente von zwei Herstellern und variieren in Größe (30cm bis 2,70m Länge) sowie Farbe (rot, gelb, schwarz, grau). Sie wurden bei wechselnden Wetterverhältnissen aufgenommen. Die Bildaufnahme erfolgte mit aus der Luft mit verschiedenen UAVs und vom Boden mit Handkameras. Die Auflösung variiert von 4000x3000 px bis 6000x4000 px. Das manuelle Labeln dieses Datensatzes dauerte rund 130 Stunden (für Beispielbilder siehe Abbildung 4a). Die Labels werden als Textdateien gespeichert und entsprechend der Anforderungen der neuronalen Netze verarbeitet.

Abbildung 4: Beispielbilder zur Klassifizierung und Erkennung

4.2 Bildanalyse

Für die Bildanalyse wird das Nvidia Deep Learning-GPU-Trainingssystem DIGITS (Yeager 2015) verwendet, das eine grafische Webschnittstelle zu den weit verbreiteten Machine-Learning-Frameworks Tensor-Flow, Caffe und Torch bietet (NVIDIA 2018). Es ermöglicht die Verwaltung der Daten, das Erstellen von CNNs und die Visualisierung des Trainingsprozesses.

4.2.1 Bildklassifizierung

Für die Klassifizierung wird ein in Caffe implementiertes GoogLeNet und der Adam Solver (Kingma und Ba 2014) verwendet. Zur Erstellung von Bildern, welche genau ein Schalelement enthalten, wurden die Baustellenfotos automatisiert entlang der Begrenzungsrechtecke beschnitten (siehe Abbildung 4b). Das Tool zur Beschneidung der Bilder wird auf GitHub als OpenSource-Lösung zur Verfügung gestellt[1]. Um relativ gleichmäßige Bilder mit ausreichender Detailgenauigkeit zu gewährleisten, wurden alle Bilder unter 200x200 px entfernt.

Um einen umfassenden Algorithmus zu trainieren, wurden zusätzliche Klassen baustellenrelevanter Objekten hinzugefügt (Tabelle 1). Die Daten aus dem Caltech 256 Datensatz benötigen keine weitere Vorbereitung (Griffin, Holub, and Perona 2007).

Tabelle 1: Klassen und Anzahl der Bilder für das Training eines Klassifizierungs-CNNs

Klasse	Herkunft	Anzahl	Klasse	Herkunft	Anzahl
Fass	Caltech 256	47	Schalung	Eigene Daten	1410
Bulldozer	Caltech 256	110	Schraubendreher	Caltech 256	102
Auto	Caltech 256	123	Schubkarre	Caltech 256	91
Stuhl	Caltech 256	62	Maulschlüssel	Caltech 256	39

In Digits wurden alle Bilder auf 256x256 px skaliert, da GoogLeNet für Bilder dieser Größe optimiert ist. DIGITS teilt die Daten automatisch in Trainings- und Validierungsdaten. Das CNN konvergiert schnell zu einer hohen Genauigkeit (Top-1-Fehler) um 85% und stagniert nach 100 Epochen bei 90% (siehe Abbildung 5). Für höhere Genauigkeiten in allen Klassen könnten zusätzliche Bilder zu den unterrepräsentierten Klassen hinzugefügt werden.

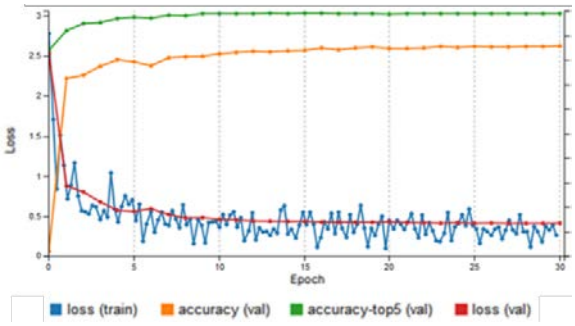

Abbildung 5: Genauigkeit (Top-1 und Top-5 Fehler) des GoogLeNets nach 30 Trainingsepochen

4.3 Objekterkennung

Um mehrere Schalungselemente in einem Baustellenbild zu erkennen, wird ein in Caffe implementiertes CNN mit DetectNet-Architektur verwendet. Um die Trainingszeit zu reduzieren, werden die Gewichte des „BVLC GoogleNet-Modells“[2] verwendet, das auf ImageNet-Daten vortrainiert wurde. Das Training wird mit dem Adam Solver durchgeführt. Für DetectNet wurden die Bilder in Patches von 1248 x 384 Pixel zerlegt und in 85% Trainings- und 15% Validierungsdaten aufgeteilt (Abbildung 4 c).

[1] https://github.com/tumcms/Labelbox2DetectNet
[2] https://github.com/NVIDIA/DIGITS/tree/master/examples/ob-ject-detection

Das CNN wurde zweimal mit jeweils 300 Epochen trainiert, wobei die zweite Trainingsrunde nur geringfügige Verbesserungen brachte. Genauigkeit und Ausfall schwanken um 65%, der mAP um 45% (Abbildung 6) – einige Elemente werden nicht oder falsch erkannt. In Abbildung 7 ist die resultierende Begrenzungsbox für ein Beispielbild dargestellt – hier wurde ein gutes Ergebnis erlangt.

Schritte zur Verbesserung der Ergebnisse erfordern eine umfangreichere Vorverarbeitung der Daten, längere Trainingsperioden und Anpassungen sowohl der Netzwerkarchitektur als auch der Lösungsalgorithmen.

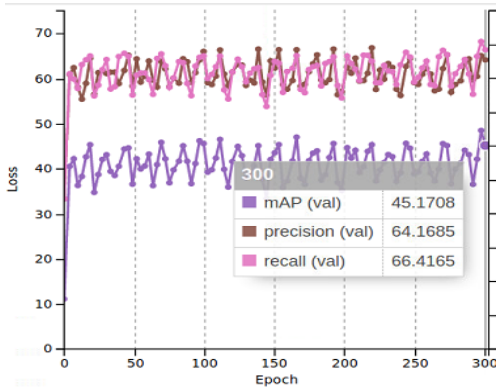

Abbildung 6: Genauigkeit, Ausfall und mAP des DetectNets nach 2 mal 300 Trainingsepochen

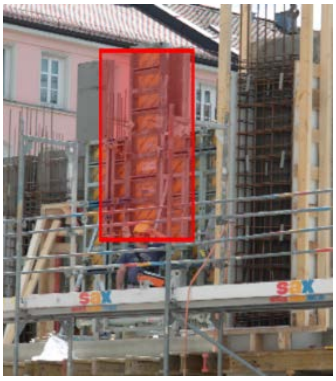

Abbildung 7: Erkanntes Begrenzungsrechteck für ein Schalungselement

5. Zusammenfassung

Dieser Beitrag konzentriert sich auf die Bildanalyse von Baustellenbildern. Um maschinelle Vorhersagen über die auf einem Bild dargestellten Konstruktionselemente zu treffen, können Algorithmen des maschinellen Lernens trainiert werden. Hier wird zunächst der aktuelle Stand der Technik im Bereich des maschinellen Lernens vorgestellt und die Algorithmen auf ihre Eignung für die Anwendung im Bereich des Bauwesens hin untersucht. Dann werden diese Ansätze auf Baustellenelementen getestet. Für das Training wurden 750 Bilder von Baustellen gelabelt, woraus knapp 10.000 beschriftete Schalungselemente resultieren. Die Bilder wurden als Eingabe für verschiedene Klassifizierungs- und Erkennungsalgorithmen verwendet, was zu sehr hohen Erfolgsraten für die Klassifizierung von Einzelobjektbildern und befriedigenden Erfolgsraten für die Objekterkennung auf Mehrobjektbildern führte. Da die Objekterkennung aktuelles Forschungsobjekt einer großen Gemeinschaft von Forschern ist, bieten die Ergebnisse einen vielversprechenden Ausgangspunkt für zukünftige Verbesserungen.

6. Danksagungen

Diese Arbeit wird von der Bayerischen Forschungsstiftung im Rahmen des Forschungsprojekts 1156-15 unterstützt. Wir danken dem Leibniz-Rechenzentrum (LRZ) der Bayerischen Akademie der Wissenschaften (BAdW) für die Unterstützung und Bereitstellung von Hochleistungsrecheninfrastrukturen, die für diese Publikation genutzt wurden.

Literatur

Albelwi, Saleh, and Ausif Mahmood. 2017. “A Framework for Designing the Architectures of Deep Convolutional Neural Networks.” *Entropy* 19 (6): 242. doi:10.3390/e19060242.

Braun, Alexander, Sebastian Tuttas, André Borrmann, and Uwe Stilla. 2015. “A Concept for Automated Construction Progress Monitoring Using BIM-Based Geometric Constraints and Photogrammetric Point Clouds.” *ITcon* 20: 68–79.

Braun, Alexander, Sebastian Tuttas, Uwe Stilla, and André Borrmann. 2018. “Process- and Computer Vision-Based Detection of as-Built Components on Construction Sites.” In *Proceedings of the 35th International Symposium on Automation and Robotics in Construction and Mining*, 7.

Buduma, Nikhil. 2017. *Fundamentals of Deep Learning : Designing Next-Generation Machine Intelligence Algorithms*. Vol. 44. doi:10.1007/s13218-012-0198-z.

Girshick, Ross. 2015. “Fast R-CNN.” In *2015 IEEE International Conference on Computer Vision (ICCV)*, 1440–48. IEEE. doi:10.1109/ICCV.2015.169.

Golparvar-fard, Mani, F Pena-Mora, and S Savarese. 2009. “D4AR - a 4 Dimensional Augmented Reality Model for Automation Construction Progress Monitoring Data Collection, Processing and Communication.” *Journal of Information Technology in Construction* 14 (June): 129–53.

Griffin, G., A. Holub, and P. Perona. 2007. “Caltech-256 Object Category Dataset.” http://www.vision.caltech.edu/Image_Datasets/Caltech256/.

Han, Kevin K., and Mani Golparvar-Fard. 2017. “Potential of Big Visual Data and Building Information Modeling for Construction Performance Analytics: An Exploratory Study.” *Automation in Construction* 73 (January): 184–98. doi:10.1016/j.autcon.2016.11.004.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. “Deep Residual Learning for Image Recognition.” In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–78. IEEE. doi:10.1109/CVPR.2016.90.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. 2017. “ImageNet Classification with Deep Convolutional Neural Networks.” *Communications of the ACM* 60 (6): 84–90. doi:10.1145/3065386.

Kropp, Christopher, Christian Koch, and Markus König. 2018. “Interior Construction State Recognition with 4D BIM Registered Image Sequences.” *Automation in Construction* 86 (February): 11–32. doi:10.1016/j.autcon.2017.10.027.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. “Deep Learning.” *Nature* 521 (7553): 436–44. doi:10.1038/nature14539.

NVIDIA. 2018. “Nvidia Digits - Deep Learning Digits Documentation,” no. May.

Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. “You Only Look Once: Unified, Real-Time Object Detection.” In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–88. IEEE. doi:10.1109/CVPR.2016.91.

Redmon, Joseph, and Ali Farhadi. 2017. “YOLO9000: Better, Faster, Stronger.” In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6517–25. IEEE. doi:10.1109/CVPR.2017.690.

Redmon, Joseph, and Ali Farhadi. 2018. “YOLOv3: An Incremental Improvement,” April.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. 2017. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (6): 1137–49. doi:10.1109/TPAMI.2016.2577031.

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, et al. 2015. “ImageNet Large Scale Visual Recognition Challenge.” *International Journal of Computer Vision* 115 (3): 211–52. doi:10.1007/s11263-015-0816-y.

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. “Going Deeper with Convolutions.” In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–9. IEEE. doi:10.1109/CVPR.2015.7298594.

Tao, Andrew, Jon Barker, and Sriya Sarathy. 2016. “DetectNet: Deep Neural Network for Object Detection in DIGITS.” https://devblogs.nvidia.com/detectnet-deep-neural-network-object-detection-digits/.

Yeager, Luke. 2015. “DIGITS : The Deep Learning GPU Training System.” *ICML AutoML Workshop*.