Technische Universität München

Ingenieurfakultät Bau Geo Umwelt

Lehrstuhl für Computergestützte Modellierung und Simulation

Automatic Classification and Consistency verification of Digital drawings using Deep Learning

Master Thesis

of the Master of Science degree in Civil Engineering

Autor: Sherief Ali

Matriculation number:

1. Supervisor: Prof. Dr.-Ing. André Borrmann

2. Supervisor: M.Sc. Jimmy Abualdenien

Date of issue: 15. April 2021

Date of Submission: 15. September 2021

Preface

The viral approach of getting everything automated has motivated construction sectors to limit the time for several tasks as the timeframe for every task is limited nowadays, cause there is an option to automatically do this task without interpretation of human beings, which might take a longer time. The incorporation of BIM and Artificial intelligence has overcome a lot of issues as well as, has processed many tasks without time consumption as the machines have performed some tasks like human beings that in former times no one predicted that one day machines can do these tasks.

There are diverse applications of Artificial intelligence, one of these applications is deep learning, which will be discussed in this study. There are several types of drawings, thus, an image classification network is implemented to classify images according to their types. Additionally, an object detection network, which works on localize the bounding boxes of the images to extract more details from the drawings like textual information

I would like to thank my supervisor Mr. Jimmy Abualdenien for the outstanding guidance and support in the whole project from day one and his patience throughout the thesis project. I would like to thank my parents and my brother for supporting me and wishing me the best of luck during my entire study at TUM. Furthermore, I would like to thank my friends for keeping me always on track. Moreover, I would like to thank TUM, which provides all the utilities and support to do your best.

15.09.2021

Sherief Ali.

Abstract

Digital drawings, which provide a detailed graphical representation, play a crucial part by featuring instant feedback on a vast amount of information among users, consequently, improving the collaboration between the team members. Artificial intelligence (AI) adopts a routine to extract features from the digital drawings and classify these drawings and as a result, a 3D model is constructed. Thus, conserving a remarkable amount of time and cost. The virtuosity of deep learning is used to allow the machines to approach human vision by enabling them to recognize and classify the images. A convolutional neural network (CNN) is a class of deep learning architectures that follows a hierarchical model to build the network, which extracts features from the images and predicts the outputs. However, a well-developed network for image classification requires a large dataset, so that the network can observe a large variety of features. In this study, two deep neural networks are proposed for the automatic classification of construction drawings and object detection using alternative strategies for feature extraction. The construction drawings are collected to form a dataset for image classification. In particular, one deep neural network is conducted to classify seven types of construction drawings. The neural network is trained using two well-known CNN architectures (VGG16, VGG19) to attain a state-of-the-art performance of the network. Diverse cases are studied by altering different hyperparameters of the network to accomplish the best possible performance of the image classifier. The second deep neural network aims to localize bounding boxes of the construction drawings and the title boxes, which provide detailed information of the drawings that are labeled. The object detection network is trained on two pre-trained models (YOLOv5m, YOLOv5s) to achieve the highest possible performance of the network. The different cases were trained, fine-tuned, and evaluated. The results showed that the deep neural network with the VGG16 architecture achieved the highest accuracy (94%) amongst the image classification neural networks and the object detection network, which used the parameters of the yolov5m model achieved an accuracy of 100% in localizing the bounding boxes and an accuracy of 93% in localizing the bounding boxes with detecting the type of the construction drawing.

Keywords: Deep Learning, Convolutional Neural Network (CNN), Object detection, Construction drawings, Image classification, VGG16, VGG19, YOLO.

Zusammenfassung

Digitale Zeichnungen, die eine detaillierte grafische Darstellung bieten, spielen eine entscheidende Rolle, da sie ein sofortiges Feedback über eine große Menge an Informationen zwischen den Nutzern ermöglichen und somit die Zusammenarbeit zwischen den Teammitgliedern verbessern. Künstliche Intelligenz (KI) extrahiert mit Hilfe einer Routine Merkmale aus den digitalen Zeichnungen und klassifiziert diese Zeichnungen, so dass ein 3D-Modell entsteht. Auf diese Weise wird eine beträchtliche Menge an Zeit und Kosten gespart. Die Virtuosität des Deep Learning wird genutzt, um die Maschinen in die Lage zu versetzen, sich dem menschlichen Sehen anzunähern, indem sie die Bilder erkennen und klassifizieren können. Ein Convolutional Neural Network (CNN) ist eine Klasse von Deep-Learning-Architekturen, die ein hierarchisches Modell zum Aufbau des Netzwerks verwenden, das Merkmale aus den Bildern extrahiert und die Ausgaben vorhersagt. Ein gut entwickeltes Netzwerk für die Bildklassifizierung erfordert jedoch einen großen Datensatz, damit das Netzwerk eine große Vielfalt an Merkmalen berücksichtigen kann. In dieser Studie werden zwei tiefe neuronale Netze für die automatische Klassifizierung von Konstruktionszeichnungen und die Objekterkennung unter Verwendung alternativer Strategien zur Merkmalsextraktion vorgeschlagen. Die Bauzeichnungen werden gesammelt, um einen Datensatz für die Bildklassifizierung zu bilden. Insbesondere wird ein tiefes neuronales Netz zur Klassifizierung von sieben Arten von Konstruktionszeichnungen eingesetzt. Das neuronale Netzwerk wird mit zwei bekannten CNN-Architekturen (VGG16, VGG19) trainiert, um eine dem Stand der Technik entsprechende Leistung des Netzwerks zu erreichen. Es werden verschiedene Fälle untersucht, indem verschiedene Hyperparameter des Netzwerks verändert werden, um die bestmögliche Leistung des Bildklassifizierers zu erreichen. Das zweite tiefe neuronale Netzwerk zielt auf die Lokalisierung von Bounding Boxes der Konstruktionszeichnungen und der Titelboxen ab, die detaillierte Informationen über die zu beschriftenden Zeichnungen liefern. Das Objekterkennungsnetz wird auf zwei vortrainierten Modellen (yolov5m, yolov5s) trainiert, um die höchstmögliche Leistung des Netzes zu erreichen. Die verschiedenen Fälle wurden trainiert, feinabgestimmt und bewertet. Die Ergebnisse zeigen, dass das tiefe neuronale Netz mit der VGG16-Architektur die höchste Genauigkeit (94 %) unter den neuronalen Netzen zur Bildklassifizierung und das Objekterkennungsnetz, das die Parameter des yolov5m-Modells verwendete, erreichten eine Genauigkeit von 100 % bei der Lokalisierung der Boundingboxen und eine Genauigkeit von 93 % bei der Lokalisierung der Boundingboxen mit Erkennung des Typs der Bauzeichnung.

Contents

List of Figures

List of Tables

List of Abbreviations

BIM	Building Information Modeling
AI	Artificial Intelligence
CNN	Convolutional Neural Network
VGG	Visual Geometry Group
YOLO	You Only Look Once
ANN	Artificial Neural Network
RNN	Recurrent Neural Network
mAP	Mean Average Precision

1 Introduction and Motivation

The initiation of Building Information Modelling (BIM) was by the American professor Charles Eastman who published his work prototype showing the maps, sections, and facades in the same document, which has a perceptive idea that any changes to any drawing alter the other drawings and thus they are all connected. Additionally, Charles Eastman eases the way for generating costs and materials. Consequently, Europe had started writing equivalent articles to grant BIM as a powerful tool in the construction industry. Recently, BIM has become the backbone of the construction industry by its remarkable tools that have shortened the time and effort needed (Tang, Shelden, Eastman, Bozorgi & Gao, 2019).

Certainly, BIM helps the construction industry in the creation of multi-dimensional models with simulation and provides an overview of the project before its commencement. The collaboration between structural, architectural, and electrical engineers has reduced the clashes that might arise during the project (Abualdenien et al., 2020). Moreover, BIM provides a real-time assessment of the model from the execution until the end of the project with simple adaptations to the model that adapts to the current status of the project. The popularity of BIM nowadays is a result of the flexibility in dealing with visualized models that reveal the faults that could arise and thus model visualization saves time, cost and secures the quality of work (Sacks, Girolami & Birlakis, 2020).

1.1 Introduction

Construction industries always seek faster and futuristic strategies that save more time and cost as compared to BIM. Despite the persistence of BIM, the perspective has to be widened ergo Artificial intelligence (AI) has been integrated into BIM. AI is a machine program that is meant to perform like the human brain. The support of BIM by AI has revealed new challenges and solved new problems that BIM might encounter. AI sectors depend on learning patterns in the data and giving outputs based on these learnings, thus, as the precision of learning increases, the precision of the outputs increases as well (Sacks, Girolami & Birlakis, 2020).

Concerning large-scale construction projects, AI plays a role in smoothing the progress of the project by implementing innovative ideas obtained while analyzing data and predicting outputs. Furthermore, AI models are developed with the help of data obtained from BIM, which contains data from previous construction projects as well as newly generated data. Nowadays, the collaboration of BIM and AI is becoming viral, since people became aware of how this collaboration might positively affect the industry in the future. The productivity of AI, the reduction of costs due to errors or clashes in construction projects, and the BIM technology has created a new era of construction and innovation (Sacks, Girolami & Birlakis, 2020).

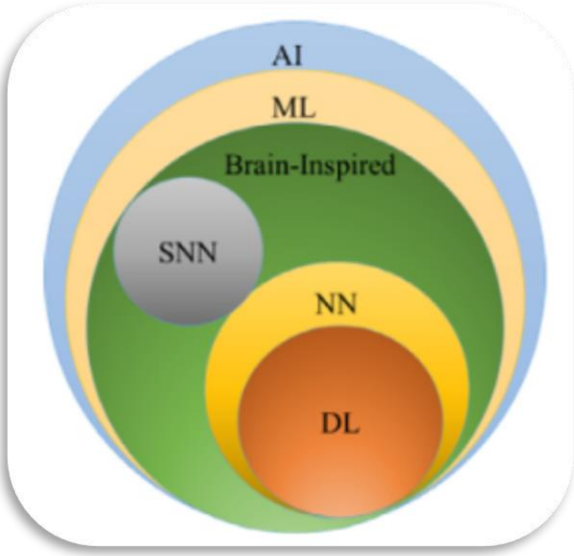


Figure 1.1 Elaboration of Artificial intelligence (Sharma, Sharma & Jindal, 2021)

Building systems were immensely simple before the introduction of AI, which had a huge impact by converting simple building systems into intelligent building systems. Machine learning has made a significant shift by supporting object detection, cloud computing, and web-based data gathering. Subsequently, machine learning has a sub-category, which is deep learning, that is specialized in specific tasks to guarantee the highest performance, for instance, sensing, object recognition, and perception (StandfordUniversity, 2016). Deep learning applies the neural networks architecture approach as the network has nodes and edges between input and output data. The remarkable deep learning network can do automatic learning of features of the inputs. The functionality of deep learning outperforms other machine learning algorithms in large and high dimensional data as a deep architecture is used to extract features from data, the process goes gradually as the initial layers learn some features from the input data and the learning of features increases from one layer to another (Sharma, Sharma & Jindal, 2021).

1.2 Motivation

The objective of image classification is the detection of common features in the images to be capable of predicting the class of the image, this can be easily done by humans but machines can’t recognize and categorize images, thus, deep learning was used to train the machines to extract features automatically from the images then predict the class that these images belong to. Therefore, this area became viral as a result of the ability to classify images without doing any effort to classify them manually, hence, deep learning is common nowadays (Corominas, Smolinska & Rana, 2021).

Apparently, the precise automated classification of images, as well as object detection that provides the ability to localize the details in the drawings, is a beneficial step for different approaches or further technical implementations such as the approach discussed by (Borrmann, Abualdenien, & Krijnen, 2021). The idea behind this research is to link the digital drawings to the BIM models as any action taken on digital drawings, the same action is shown on the BIM model as well. Primarily, the research purpose is to provide an interactive environment, accordingly, the digital drawings linked to BIM models need to be classified according to their types to eliminate the errors or clashes that might arise during linking both of them. (Borrmann, Abualdenien, & Krijnen, 2021) has discussed that BIM models mightn’t be enough as there are some details in the 2D drawings that have to be considered. For instance, there are specific details in 2D drawings such as dimensions, symbols, or might be particular information in the title boxes that isn’t illustrated in the drawings extracted from BIM models. Hence, the information from 2D drawings can be extracted by object detection through localizing the information, so that the information and the text can be extracted.

.This thesis presents a technique of automatic classification of digital drawings, our concern is the construction drawings and there are many types, thus, seven types of construction drawings are chosen to be implemented in the dataset. The objective is to train the machine on the provided dataset to have the ability to predict the classification to which each image belongs. The automatic classification of images can solve the issue of time consumption that is needed to classify images of a large project, for instance, plenty of construction drawings of a shopping mall or a train station that need to be classified according to their type, thus, by using our method, the images of mega projects can be classified accordingly in less time.

The method used has two folds. First, training a neural network to extract features from the construction drawings then classifying the images accordingly, to achieve the highest possible precision. Diverse drawings are needed in the dataset to give the network the ability to learn and extract features from each class of the images. Second, the accuracy is our concern, thus, training another neural network to detect the objects in the drawing, the objects that we target to detect are the title blocks presented in the drawings, which show the type of each construction drawing. Consequently, providing the network a dataset of labeled images then predicting the objects automatically, therefore, the network can localize the bounding boxes.

1.3 Structure of the thesis

This section defines briefly the structure of the thesis to provide an overview of the content of the following chapters

- Chapter 2 discusses the theoretical background of topics related to the vision of the thesis as well as, the related research journals and papers discussed the area of this research before. In particular, the influence of bim implementation on construction sites is discussed with an overview of types of neural networks available and a brief of machine learning and deep learning; Furthermore, an overview of convolutional neural networks (CNN) and their architectures. Additionally, The hyperparameters, which influence the performance of the neural network, are discussed as well. Some examples of construction drawings are illustrated and examples of construction drawings symbols and thus the usability of digital drawings and how they can be extracted from 3D models are discussed clearly. Finally, there is an overview of real-time object detection (YOLO).

- Chapter 3 explains the methodology used to come out with this thesis study by defining the methods used and how the data is collected for the neural networks implemented in this study.

- Chapter 4 gives an overview of the image classification network implemented as well as the neural network structure and the two CNN architectures used (VGG16 and VGG19).

- Chapter 5 explains the approaches used to reach the highest performance possible for an image classification network and a detailed description of the variables changed in every approach.

- Chapter 6 defines the object detection neural network with an illustration of the architecture of the network

- Chapter 7 explains the approaches used in the object detection neural network and a detailed description of each trial used regarding inputs and variables.

- Chapter 8 summarizes the results of the neural networks implemented and their approaches as well as the study limitation are discussed.

- Chapter 9 concludes the thesis study and defines some aspects for future work

2 Theoretical Background and Related Works

In the following sections, a general theoretical background of the information needed for project implementation is presented. The influence of bim on construction sites, and how BIM has changed the progress of work is illustrated in Section 2.1. There are diverse types of construction drawings, thus, the types of drawings, examples, symbols on the drawings, and how the digital drawings can be extracted from 3D models, are discussed in Section 2.2. Moreover, Section 2.3 gives an overview of neural networks and their types. Additionally, in section 2.3.4 there is a comparison between types of neural networks, in comparison to Section 2.4, which defines machine learning and deep learning. Some examples of CNN architectures are discussed and a comparison between VGG16 and VGG19 is provided in section 2.5. The hyperparameters, which influence the performance of neural networks, are discussed in Section 2.6. Section 2.7 provides an overview of real-time object detection and some samples of how object detection works. Finally, Section 2.8 provides an overview of research papers related to this study.

2.1 BIM influence on Construction projects

Building Information Modelling (BIM) is the methodology that has shifted former construction projects into a more futuristic and trackable one. The adoption of BIM took almost two decades to reach this advancement as well as, for knowing how vigorous and beneficial BIM is. The benefit of BIM is work collaboration between engineers (mechanical, electrical, and Structural), architects, and consultants, all have the same objective to enhance the performance of work (Bargstadt, 2015).

The hallmarks of BIM are the 3D representation of buildings and facilities not only a three-dimensional representation but also BIM could reach seven-dimensional models. Thus, incrementation in dimensions increases various features. Into the bargain, tracking and supervising the project from execution until reaching the final stage is accessible, by which issues can be recognized and revealed from the different views available (Abualdenien & Borrmann, 2019). Disclosing one of the facilities of BIM, the five-dimensional models, which includes the 3D model, schedule, and cost of the project. A 5D model is a powerful tool that enhances the progress of construction work as the necessary details are monitored before launching the project; furthermore, costs can be adapted according to the market prices alterations (Bargstadt, 2015).

Despite the metamorphosis BIM has created, the binding contracts are paper-based until now, and even if it is documented electronically, it is mandatory to be printed out to complete the process of approval (Abualdenien & Borrmann, 2020). Consequently, this results in a misuse of the BIM feature as construction associations have defectiveness in the data provided by BIM models, which hasn't gathered all project data in one hand (Bargstadt, 2015).

2.2 Construction Drawings

The main fragment of the construction process is construction drawings, hence, the construction drawings give an overview of the whole project revealing the data that the project needs to be commenced. Moreover, detailed construction drawings reveal the quantities needed for every division, for instance, the number of floors available or the design of MEP. In former times, construction 2d drawings were in paper-based format only, while the drawings nowadays are available digitally and paper-based as well. The construction drawings are the commencement point for any construction project, but the format of the drawing used differs from one project to another. The responsibility of handing over the construction drawings doesn't rely on one department, the construction drawings run through long procedures of discussion, approval, and modifications to guarantee that these drawings are precise and can be conveyed to the construction process. There are numerous types of construction drawings such as Elevation, Section, Floor plans, Electrical, etc. Figure 2.9 and Figure 2.10 demonstrate two examples of construction drawings available (Manrique, Hussein, Bouferguene & Nasseri, 2015).

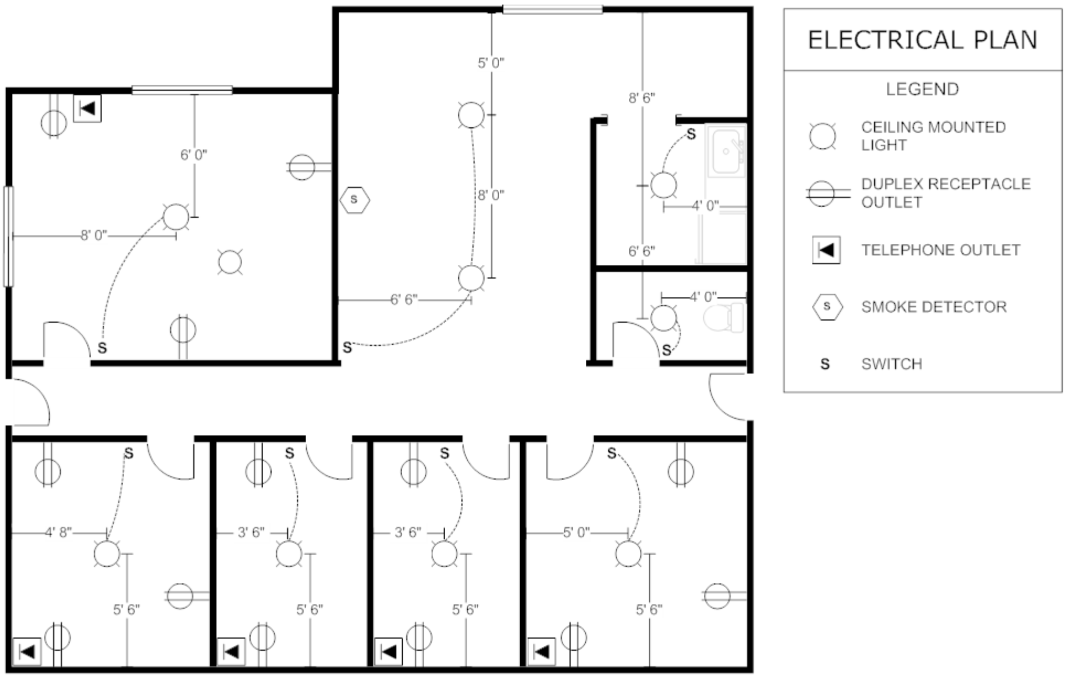


Figure 2.1 Electrical Plan (Sample from the Dataset)

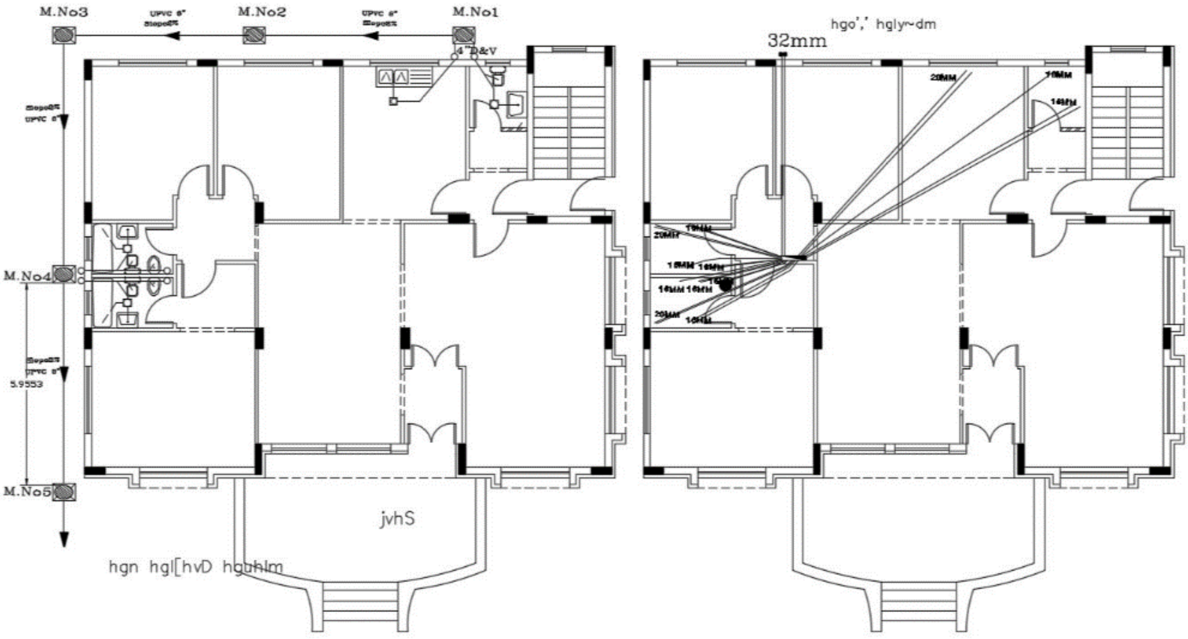


Figure 2.2 Plumbing Plan (Sample from the Dataset)

2.2.1 Symbols of Construction Drawings

The construction 2D drawings have diverse symbols, hence, the reader can know and read several things from these symbols. These symbols are drawn by the structural or architectural engineer. For instance, the 2D drawings have special lines for every constraint, textual information, dimensions, etc. There is also a special representation of doors and windows in the drawings, to show the type and the orientation of the item. Furthermore, the thickness of the lines indicates the type of the wall and whether it is a concrete wall or a brick wall. By the way of illustration, Figure 2.11 shows several symbols that are drawn on construction drawings, and thus the textual information is clearly shown, which defines boundaries for rooms. The orientation of the doors and the thickness of the lines are shown as well. Therefore, the symbols have a main role in the 2D drawings to provide precise information for the drawing reader (Rezvanifar, Cote, & Albu, 2019).

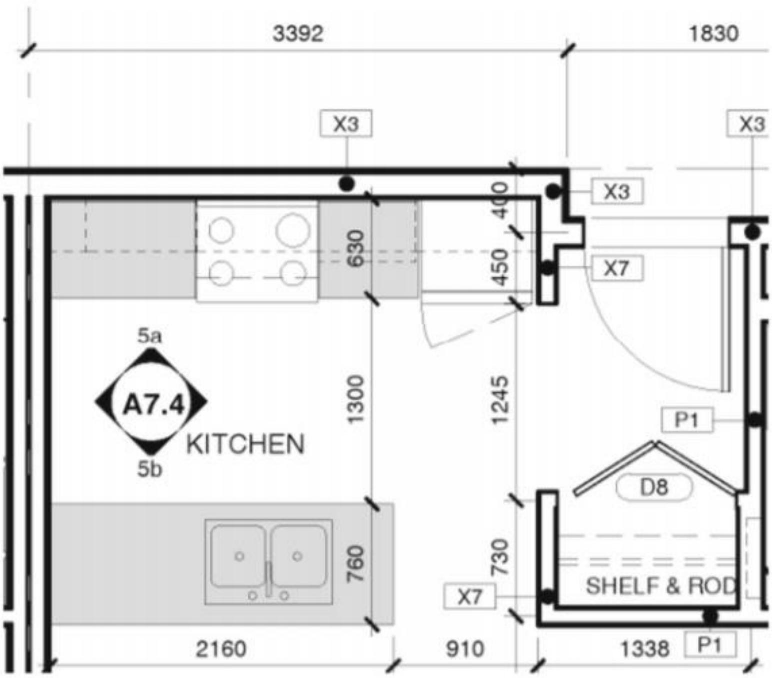


Figure 2.3 Symbols of construction drawings (Rezvanifar, Cote, & Albu, 2019)

2.2.2 Digital Drawings

There are several sources to get 2D drawings, one of these sources is the 3D models. Several software programs, which are responsible for producing a 3D model for construction projects, one of these software programs is Autodesk Revit. In particular, Revit has two options regarding extracting of forming 2D construction drawings, the first one is complete detailing and drawing the 2D drawing on Revit and thus adding all the specifications and information to the drawing, thereafter, these 2D drawings can be exported in several formats to be used in different works. While the other option is importing a full detailed 2D drawing into the software and then doing some configurations on it. Therefore, the 3D model can be formed easily as the 2D plan has been taken as a reference. Figure 2.12 shows the first option discussed previously as this is an example of a complete 3D model, which is used to extract some 2D drawings, for instance, elevation plans and floor plans.

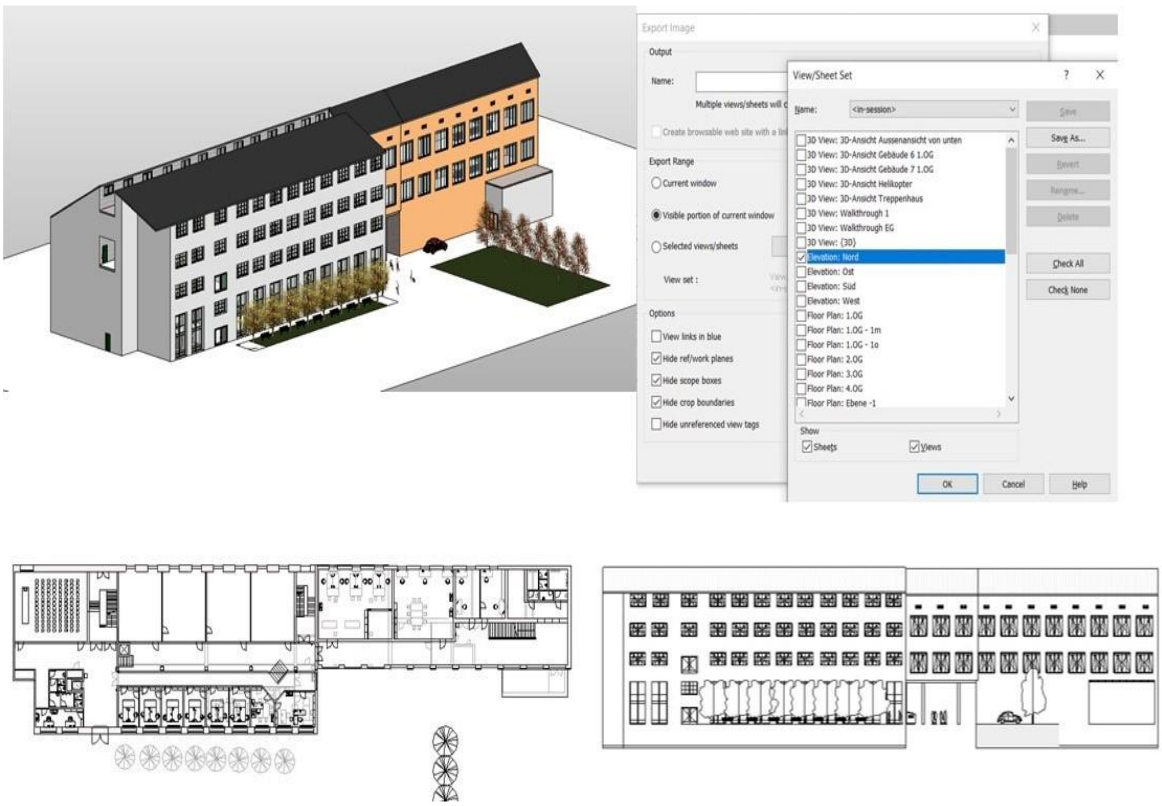


Figure 2.4 The possibility of extracting 2D drawings from a 3D model (Extracted from Revit)

2.3 Neural Networks

A neural network typically works like the human brain through identifying and recognizing the features and relationships between different sets of data. The act of neural networks is defined by a set of neurons that can ameliorate over time to achieve the fittest results amalgamating the best outputs (Chein, 2020). There are a diversity of types of neural networks, three types of them are discussed clearly in the next sections, and in the last section, there is a table showing a comparison between the three different types in the bargain.

2.3.1 Artificial Neural Network (ANN)

Artificial Neural Network as noted by Feed Forward Neural Network is a comprise of neurons that are available at each layer. ANN is known by Feed Forward as it is characterized by the flow of inputs in the forward direction only. ANN has been named as Universal Function Approximators, by virtue of its ability to learn any nonlinear functions, additionally, the capability of ANN to learn weights of any capacity (Pai, 2020).

Figure 2.1 shows the layers of ANN, it distinctly consists of three layers (input, hidden, and output) and each layer has a role, where each layer recognizes specific weights as the input layer deals with the inputs to the network, while the hidden layer processes the inputs. On the other hand, the output layer is responsible for giving out the results (Pai, 2020).

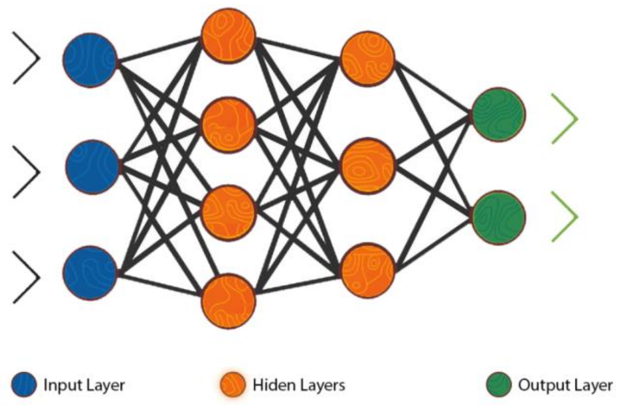


Figure 2.5 Layers of ANN (Pai, 2020)

Nonetheless, ANN encounters a struggle in image classification as if image classification is performed by ANN, some procedures needed to be performed before, for instance, the dataset of 2D images has to be converted first to one-dimensional images, therefore, ANN can have the ability to train the model (Pai, 2020).

2.3.2 Recurrent Neural Network (RNN)

The weights might have a short or long-term memory, thus, neural networks might have short or long-term memory. Accordingly, RNN is a short-term memory for its proficiency in memorizing only things that happened in the previous observations but with limited memory (Zaremba, 2015). RNN has the capability of dealing with sequential data, for instance, speech recognition, Text, and Natural Language Processing (Goodfellow, Bengio & Courville, 2017).

RNN characteristics in the processing of inputs are equivalent to Feed-Forward neural networks mentioned in the former section, as the processing of inputs in both of them depends if inputs are sequentially dependent or not on the point that RNN can process multiple inputs, In the same manner, having only the condition not sequentially dependent. In other words, RNN can process multiple elements without a sequence or two separate sequences at the same time (Goodfellow, Bengio & Courville, 2017).

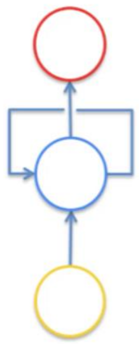


Figure 2.6 Input, Hidden, and Output layers of RNN (Zaremba, 2015).

Figure 2.2 illustrates the idea behind RNN and describing short-term memory. The yellow circle is the input, the blue circle is the hidden layer and the red circle is the output The figure reveals the role of the hidden layer as the hidden layer performs two actions, the first one is giving the data to output, while the second main role is that the hidden layer returns the data into the hidden layer back again (Zaremba, 2015).

On the grounds that, RNN and ANN have some common features that might make a misconception of understanding their roles. Figure 2.3 illustrates a comparison in differences between them as shown on the left it is visible how the hidden layers act with the data, as it proceeds data to output and proceeds data to itself again, on the other hand, the hidden layer in Feed Forward Neural Network on the right shows the hidden layer with only one act, which is the processing of data to outputs only without further processing of data to itself again (Pai, 2020).

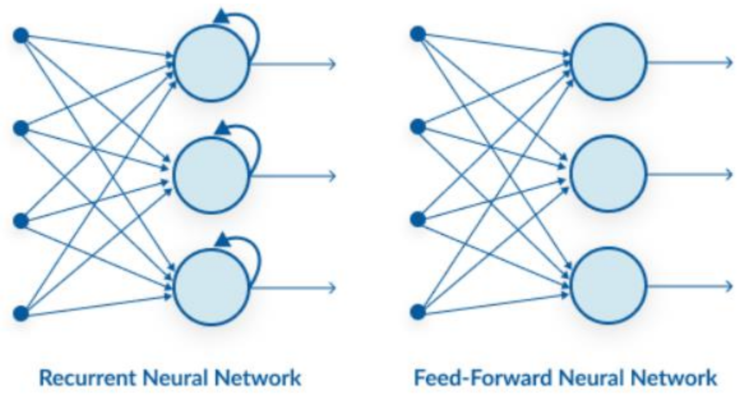


Figure 2.7 Difference between RNN and ANN (Pai, 2020)

2.3.3 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) consists of two main parts, the first part is several different convolutional and pooling layers for extracting features and the other part is fully connected layers that work on the extracted features to be able to do classification (Stabinger, Peer, and Sanchez, 2021).

Three components form a convolutional layer, these components are linear mapping, activation function, and pooling function. Succinctly, the activation function brings the nonlinear mapping feature into the network, while pooling uses the statistical characteristic of the local region to be able to give a representation of the output of a certain position. To clarify pooling, it can be adjusted whether max pooling, min pooling, or average pooling, and they can adjust noise interference in images to have a clear image to be able to reserve more features (Jia et al., 2021).

CNN works in stages as the neurons of CNN are activated by the encoder, which extracts some features which run the first layers of the network. The second stage is the act of preprocessors, which extract and compresses information to reinforce the receptive field by two ways pooling or convolution’s strides. The last stage is the turn of fully connected layers, as they now have a reinforced receptive field that can be detected and have the ability to extract the information from convolutional layer (Stabinger, Peer, and Sanchez, 2021).

Figure 2.4 illustrates the structure of the convolutional neural network.

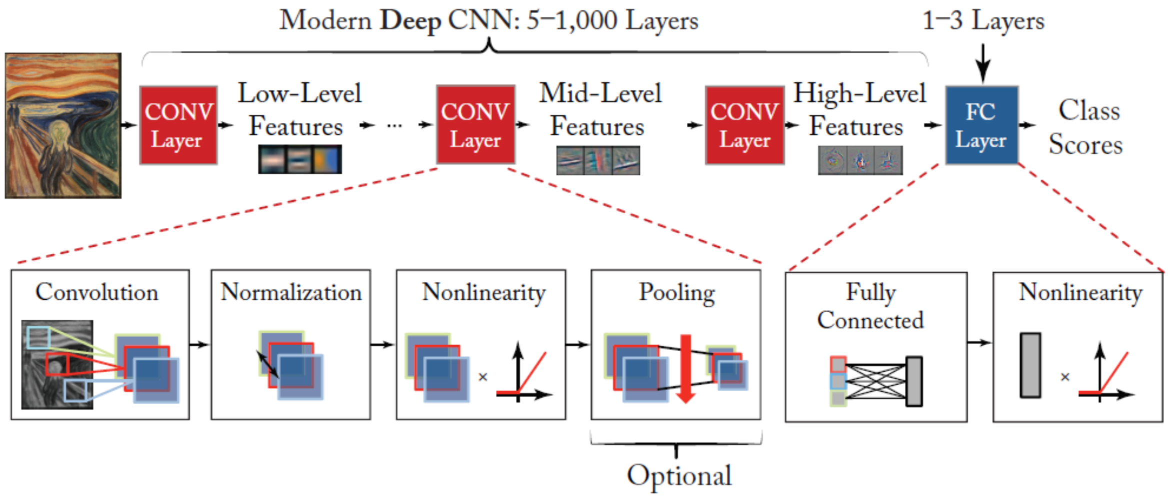


Figure 2.8 Convolutional Neural Network (Sze, 2017)

2.3.4 Comparison between neural networks types

Recapitulating the types of neural networks argued before. Table 2.1 shows a comparison between the neural networks; moreover, the type of data that can be processed to attain the best results.

	ANN	RNN	CNN
Data	Tabular Data	Sequence data	Image data
Recurrent Connections	✗	✓	✗
Parameter sharing	✗	✓	✓
Spatial relationship	✗	✗	✓
Vanishing and Exploding Gradient	✓	✓	✓

Table 2.1 Comparison between ANN, RNN, and CNN (Pai, 2020)

2.4 Machine Learning and Deep Learning

2.4.1 Machine Learning

Machine learning is the performance of a computer whose performance gets better progressively as a consequence of gaining experience from information provided or common features in the data provided. Hence, machine learning can undertake rational tasks, for instance, object detection and natural language translation. The central concept of machine learning is algorithms that are applied to a specific training dataset, therefore, training data to find common features, information, and complex patterns. Accordingly, the performance of machine learning in the classification of images, clustering, and regression can reach high levels due to the pertinency of machine learning (Janiesch, Zschech & Heinrich, 2021).

Consequently, machine learning can be applicable in many areas and already has been applied in fraud detection, credit scoring, next-best offer analysis, speech and image recognition, or natural language processing. Machine learning can provide virtuous results, nevertheless, the choice of machine learning type to use is the main factor as machine learning has three types named supervised learning, unsupervised learning, and reinforcement learning. Each type guarantee high performance in the task that is applicable to be performed by this type, for instance, supervised learning is used in applications of electronic markets and forecast of stock markets to have the ability to detect customer needs, while unsupervised learning can do market segmentation using customer reviews and reinforcement learning can do market making. The next sections give an overview of each type of machine learning and the idea behind each one of them (Janiesch, Zschech & Heinrich, 2021).

2.4.1.1 Supervised learning

The practicality of supervised learning is clear from its name as it is learning based on supervision by employing a training dataset enclosing various inputs in conjunction with labeled data or sometimes accompanied with target values to get outputs. The training dataset has inputs and outputs which are used to calibrate the parameters of the model. The procedures are initiated by training the model and then trying to predict the target values (Janiesch, Zschech & Heinrich, 2021).

2.4.1.2 Unsupervised learning

On the contrary, unsupervised learning works with a training dataset that has various inputs but without having labeled data or target values to have the ability to provide an output, therefore, the training dataset is trained to have a target of detecting points of interest and learning features or information from different inputs having shared properties (Janiesch, Zschech & Heinrich, 2021).

2.4.1.3 Reinforcement Learning

Regarding the Reinforcement learning system, it doesn't rely on the provision of inputs and outputs like supervised learning or unsupervised learning, while reinforcement learning relies on defining a goal as well as the state of the system, thus, adding specific actions that are allowed and the environmental constraints that allow providing outcomes (Janiesch, Zschech & Heinrich, 2021).

2.4.2 Deep Learning

Deep learning is a subset of machine learning. Deep neural networks have more than one hidden layer where they are arranged in deep network architectures. Deep neural networks have the capability of using multiple activations in one neuron, thus, deep neural have the ability to deal with raw input data and figure out the features and information from this data (Janiesch, Zschech & Heinrich, 2021).

There is a term called shallow machine learning like supervised learning that is discussed in the previous section, as it is one of the types of shallow machine learning. Deep learning can deal with large and high dimensional data, hence, the performance of deep learning with text, image, video, speech, and audio data is higher than shallow machine learning. Consequently, shallow machine learning performs properly with a low-dimensional set of data (Janiesch, Zschech & Heinrich, 2021).

AI is a branch of computer science, which relies on theories, methods, techniques, and applications, which aim to reach human intelligence. The objective of artificial intelligence is to provide a machine that can respond and accept orders much like the intelligence of a human. As mentioned before deep learning is a subset of machine learning, as well as machine learning, which is a branch of Artificial intelligence (Xin et al., 2018).

To sum up, Figure 2.5 illustrates the networks of both machine learning and deep learning.

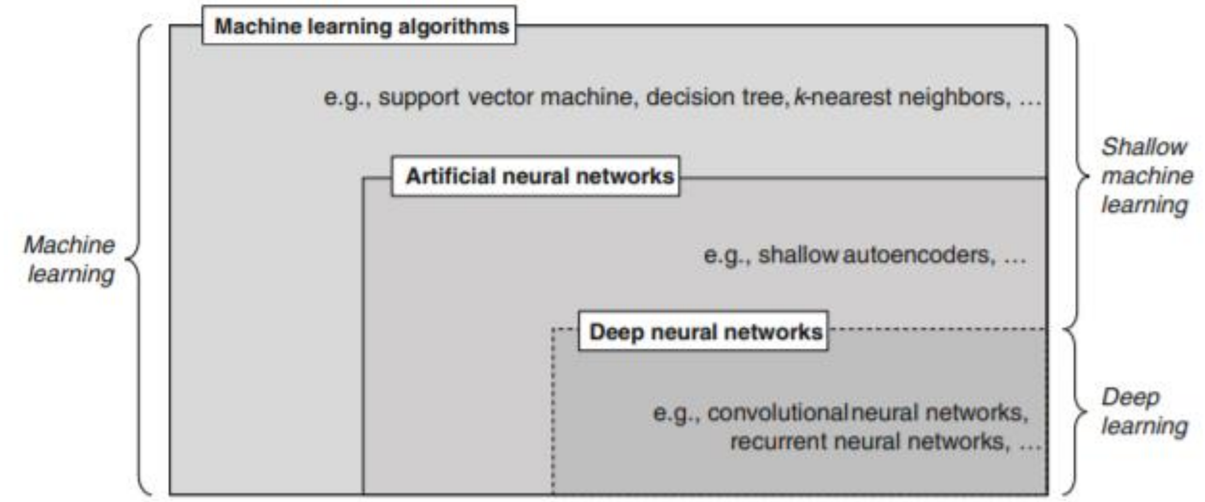


Figure 2.9 illustration of Machine learning and Deep learning (Janiesch, Zschech & Heinrich, 2021).

2.5 Convolutional neural networks architectures

The neural network has layers that are trained, the initial layers of the network can identify a small number of features from the images. As the neural network goes further, the features that can be extracted are increased till reaching the last layer of the neural network, which can classify the image based on the image content (Brodzicki, Korjakowaska, Kleczek, Garland & Bogyo, 2020).

The idea behind pre-trained models is that there are some models, which are pre-trained before, therefore, they have previously extracted many features for large datasets of images and can detect different information from images. The usage of pre-trained models converges upon the last layer of the pre-trained model as this layer would be changed to a layer that fits your model, hence, taking the leverage from all features extracted by the previous layers (Brodzicki, Korjakowaska, Kleczek, Garland & Bogyo, 2020).

Accordingly, there are two well-known pretrained models available, which are VGG16 and VGG19. VGG stands for Visual Geometry Group introduced by Simonyan and Zisserman in 2014. The numbers beside VGG indicate the number of weight layers and thus VGG16 has 16 weight layers (Rosebrock, 2017).

VGG network has 3x3 convolutional layers settled at top of each other and the depth is increasing gradually. Additionally, VGG has two fully connected layers with a total of 4096 nodes ended by Softmax classifier (Rosebrock, 2017).

There are some differences between VGG16 and VGG19, for instance, the size of the layer, filter sizes. Accordingly, Table 2.2 shows a comparison between the two architectures.

Layer	**VGG16**	**VGG19**
Size of layer	41	47
Image input size	224 x 224 Pixel	224 x 224 Pixel
Convolutional Layer	13	16
Filter size	64 & 128	64,128,256, & 512
ReLU	5	18
Max Pooling	5	5
FCL	3	3
Dropout	0.5	0.5
Softmax	1	1

Table 2.2 Comparison between VGG16 and VGG19 (Setiawan & Damayanti, 2019)

2.6 Neural Network performance

2.6.1 Batch Size

Several parameters influence the neural network performance, in other words, influence the learning of neural networks to the dataset. The batch size is one of the most important hyperparameters, which influence the learning of the network. The neural networks are trained using gradient descent and the weights are updated using the estimate of the error. The batch size is briefly the number of examples in the training dataset, which are used to estimate the error gradient. The training is influenced by the training dataset as the more dataset available for training, the more accurate the estimate of the training and the performance of the model gets better. There are three main approaches for gradient descent and every approach provides an appropriate performance of the network according to the type of the network and the dataset as well (Brownlee, 2020).

The three main types of gradient descent are

- Batch gradient descent
- Stochastic gradient descent
- Minibatch gradient descent

The types of gradient descent differ in their approaches and sizes as well as batch gradient descent size is set to the number of examples in the dataset, while stochastic gradient descent, the batch size value is set to 1, on the other hand, the minibatch gradient descent size is set to more than or less than the total number of the training dataset. To clarify the idea of batch size, the batch size of 32 defines that there are 32 samples from the training dataset that are used to get the error gradient of the network. The performance of the neural network can be altered by several parameters and the batch size is a major parameter, which its size is determined according to the size of the dataset. The batch gradient descent type is not commonly used because it sets the batch size to the total number of training datasets, hence, the value of the batch size will be very high and the performance of the model decreases gradually as the batch size increases up to a certain point as a result of several reasons, which are the usage of small batch size gives a regularization effect to the model and decreases the generalization error. Additionally, the higher the batch size, the more memory is needed to train the data, therefore, the training process becomes more difficult if the batch size is set to high values (Brownlee, 2020).

2.6.2 Learning rate

The other hyperparameter that influences the performance of the model is the learning rate. Learning rate describes the amount of change to the model in every step during training the model. The learning rate plays an important role in tuning the neural network and thus improves the capability of the network to extract features or predict the output. The learning rate values are adjusted according to the performance of the network as the learning rate varies from 0.0 to 1.0. Therefore, there is no specific number to set the learning rate as it depends on the performance of the network, size of the dataset, etc. In particular, the learning rate with a value of 0.1, reveals that the weights in the neural network are updated 0.1x estimated weight error. Accordingly, the learning rate role is to monitor the rate of learning in the model and thus the larger the learning rate, the faster the model learns and the faster the training duration. However, the fast learning of the model is not a good indication of a good performance as the accuracy of learning might drop out. On the other hand, a small learning rate takes a longer time in training and learning, while the prediction of the network might be precise. Consequently, the adjustment of the learning rate is mandatory as an optimum value has to be chosen to get the best performance possible from the network (Brownlee, 2020).

2.6.3 Batch Normalization

Batch normalization is a hyperparameter that can be implemented in almost all types of networks. The benefit of implementing batch normalization parameters in a network is to normalize the inputs to a layer. There are several ways to implement the batch normalization parameter as it can be implemented before or after the activation function. The idea behind the batch normalization parameter is to ensure the stability of distribution during training the network. As previously mentioned that the learning rate influences the speed of the training, the batch normalization parameter influences the speed of training as well (Brownlee, 2020).

2.6.4 Regularization

The main concern of training a neural network is to extract features during the process and not memorizing only the input data, hence, the generalization has to be maintained during training. There are different situations during training the neural network reveals the situation of the model and if the model is learning precisely or not. In particular, there are three types of model fitting, which are overfitting, underfitting, and good fitting. The situation of an overfitting model as in Figure 2.6 indicates that the model is performing well with the training dataset and the performance is inaccurate regarding the validation dataset. On the other hand, the underfitting model means a bad performance in both the training dataset and validation dataset as shown in Figure 2.7. On the contrary, a good fit model specifications are a good performance in learning the training dataset and the model can generalize the learned features, thus, these features can be used with any dataset. An illustration for a good fit model is shown in Figure 2.8.

Consequently, the model has to be monitored during training, therefore, the regularization parameter is the most important in controlling the overfitting and underfitting issue (Brownlee, 2020).

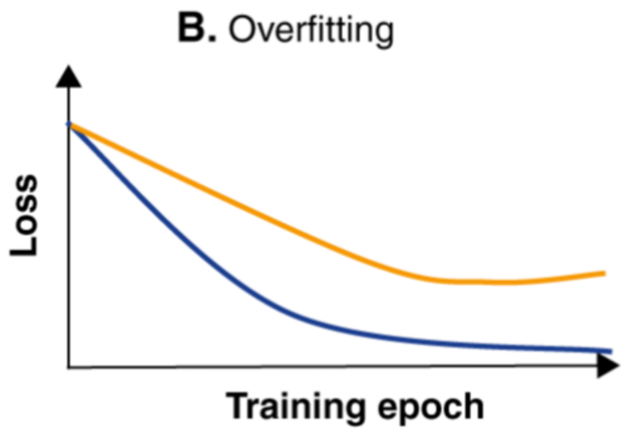


Figure 2.10 Representation for an overfitting model (Cai et al, 2020)

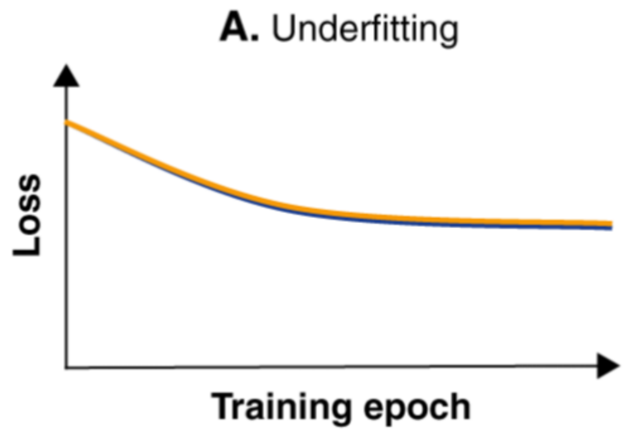


Figure 2.11 Representation for an underfitting model (Cai et al, 2020)

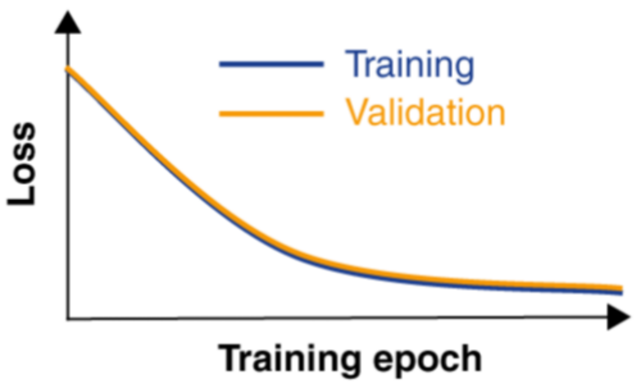


Figure 2.12 Representation for a Good fit model (Cai et al, 2020)

2.7 Real-time Object Detection YOLO

Object detection is a field of computer vision that specializes in detection and object classification, thus, object detection is capable of detection and recognition of faces. The idea behind object detection is locating bounding boxes and these bounding boxes are annotated to the dataset of images. Hence, the dataset is annotated to be implemented in the network, which localizes the labeled data.

You Only Look Once (YOLO) is capable of detecting and classifying images in only one step as the prediction of bounding boxes can be processed for the image after executing only one evaluation. The remarkable thing in YOLO is coincidental bounding box predictions and predictions of classes. The procedures of YOLO start by dividing the input image into a grid then the bounding boxes that are annotated to the image is being defined in every grid, giving a confidence score, which is an indication of the probability of the existence of a bounding box in the grid (Huang, Pedoeem & Chen, 2018).

$$C = \Pr(Object) * IOU(truth\ pred) \tag{2.1}$$

This formula illustrates how the confidence score is calculated as IOU is the intersection over union, which is a fraction from 0 to 1 where 0 defines no intersection and 1 means intersection of overlapping area between the predicted bounding box and ground truth, while the union is the area of both predicted bounding box and ground truth. The best-case scenario is an IOU of 1 or close to it as this indicates that the prediction is similar to the ground truth. There is an illustration in Figure 2.13 representing the idea of intersection and union (Huang, Pedoeem & Chen, 2018).

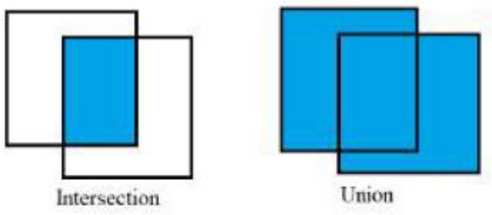


Figure 2.13 Representation of Intersection and Union (Huang, Pedoeem & Chen, 2018)

The grid cells are responsible for predicting conditional class probability, which is calculated as follows:

$$\Pr(Class_i \mid Object) * \Pr(Object) * IOU(truth\,pred) = \Pr(Class_i) * IOU(truth\,pred) \tag{2.2}$$

Regarding the calculation of loss and optimizing the confidence, the below equation is used

$$\begin{aligned} Loss = {} & \gamma_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{A} 1_{ij}^{obj} \left[(b_{xi} - b_{x'i})^2 + \left(b_{yi} - b_{y'i}\right)^2\right] \\ & + \gamma_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{A} 1_{ij}^{obj} [\left(\sqrt{b_{wi}} - \sqrt{b_{w'i}}\right)^2 + \left(\sqrt{b_{hi}} - \sqrt{b_{h'i}}\right)^2 \\ & + \sum_{i=0}^{s^2} \sum_{j=0}^{A} 1_{ij}^{obj} [(C_i - C'_i)^2 + \gamma_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^{A} 1_{ij}^{noobj} [(C_i - C'_i)^2 \\ & + \sum_{i=0}^{s^2} 1_i^{obj} \sum_{c \in classes} (P_i(c) - P'_i(c))^2 \end{aligned} \tag{2.3}$$

The loss function is used to correct the center and the bounding box of each prediction. The main role of the loss function is to correct the center and bounding box of predictions. b_x and b_y, which are indications for the center of prediction. On the other hand, bounding box dimensions are b_w and b_h. While γ_{coord} and γ_{noobj} are responsible for increasing or decreasing emphasis on bounding boxes. Moreover, C is the confidence and $P(c)$ refers to classification prediction (Huang, Pedoeem & Chen, 2018).

Figure 2.14 and Figure 2.15 show some examples of how object detection works.

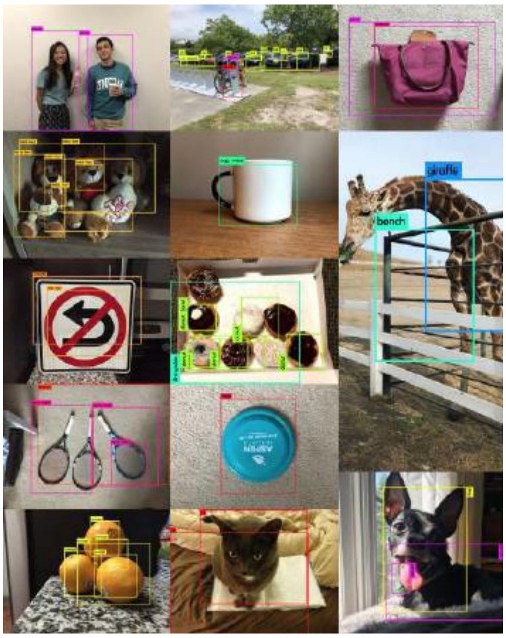


Figure 2.14 Examples of object Detection (Huang, Pedoeem & Chen, 2018)

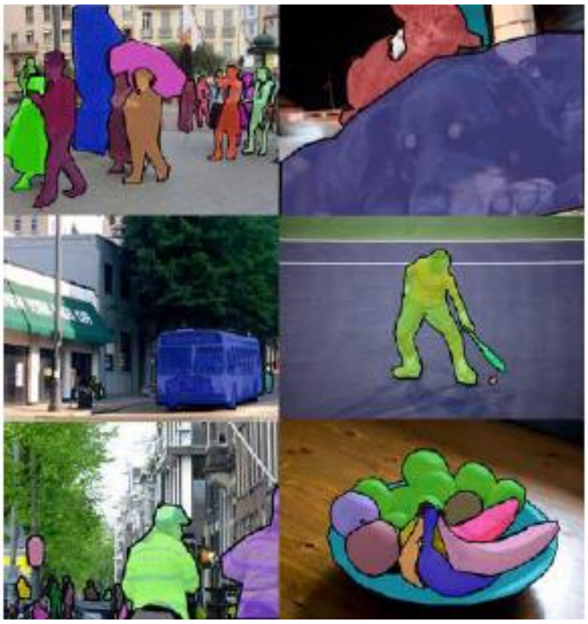


Figure 2.15 Examples of image segmentation (Huang, Pedoeem & Chen, 2018)

2.8 Related Work

This section imparts an overview of related work to different techniques and comparable ideas used in this paper.

Primarily, (Zeng, Li, Yu & Fu, 2019) applied a method to recognize floor plan elements, these elements are walls, doors, windows, and room regions. Additionally, the objective is to have the ability to recognize types of rooms available in the floor plans, for instance, rooms like dining room, bedroom, and bathroom, etc. The basic notion is setting the floor plans in a hierarchy, therefore, the pixels of floor plans are identified and categorized into inside and outside pixels as the room boundary pixels are walls, windows, and doors. Accordingly, a deep multitask network is established to detect room boundary elements; moreover, predicting room type elements. This study has reached an overall accuracy of 89%, which is not high enough to predict room boundaries of floor plans accurately and that was due to several parameters as the network struggled in recognizing different room structures such as differentiating inside and outside regions, additionally, the wall elements haven't been recognized accurately and these issues were due to the lack of data that the network is trained on, thus, the network has been trained precisely on some orientations of floor plans but not on the other orientations. Figure 2.16 illustrates how the floor plan elements are recognized in training the neural network.

Figure 2.16 Floor plan recognition (Zeng, Li, Yu & Fu, 2019)

(Ahmed, Liwicki, Weber & Dengel, 2011) has worked on improvements in automatic analysis of architectural floor plans by applying segmentation algorithms to have the ability to extract information, consequently, the extracted information is analyzed to get the structure of the rooms of the floor plans; moreover, the functions of the rooms are detected by semantic analysis. The network has been trained on 80-floor plan images and the size of the dataset has limited the accuracy reached by this network as the accuracy achieved is 89%, which is not a precise accuracy for generalization of features extraction from floor plans. Figure 2.17 shows a floor plan image and how the room boundaries are detected in the network.

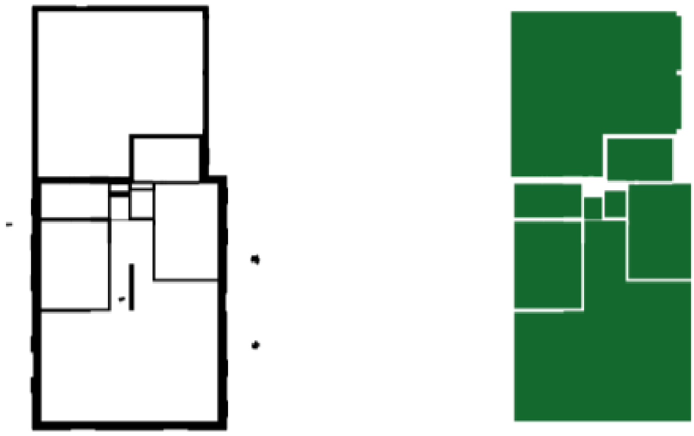


Figure 2.17 Illustration of room detection for the floor plan (Ahmed, Liwicki, Weber & Dengel, 2011)

(Simonyan & Zisserman, 2015) established very deep convolutional neural networks to have the capability to recognize large-scale images. Accordingly, alterations have been done to the layers of convolutional neural networks to achieve the highest performance possible for large-scale images, hence, incrementing the depth of layers weight to reach 16 to 19. Therefore, this paper has found an architecture of ConvNet that performs well with other datasets as well, accordingly, other datasets can be recognized with this architecture securing superb performance.

Step up a notch to (Or, Wong, Yu & Chang, 2005) by converting 2D floor plan images into 3D models. Consequently, the floor plans are characterized by walls of different thicknesses, and pairs of thin lines, which are an indication of the openings. Hence the 3D floor plan images are parsed into diverse connected segments as well as finding out the relationship to have the capability of generating models. The recognition of 2D floor plans has some conflicts which have been revealed during generating 3D model and these conflicts are extra added parts to the floor plans, additionally, the detection of the doors wasn't accurate enough as not all the doors are detected precisely, as there are some doors in the recognition are missed up. To illustrate the idea of the study, Figure 2.18 illustrates the conversion of a 2D floor plan to a 3D model.

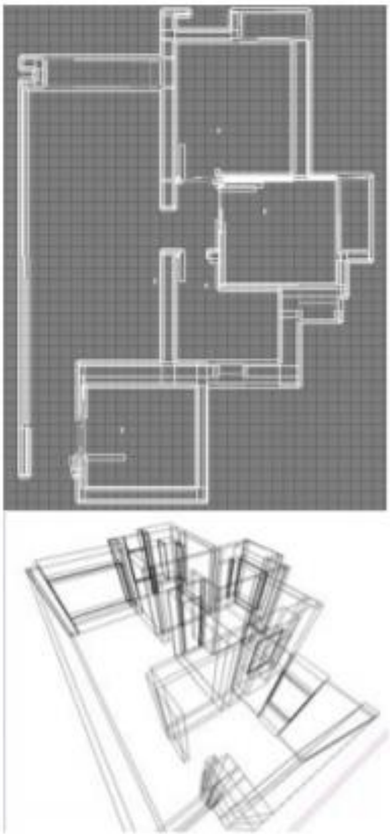


Figure 2.18 Conversion of the 2D floor plan into a 3D model (Or, Wong, Yu & Chang, 2005)

Finally, the purpose of our method is to automatically classify digital drawings using deep learning. Most of the previous work discussed before they haven’t tackled the classification of different types of construction drawings as the main concern was to recognize or detect features from a specific type of drawings, not from diverse types of drawings and classify them according to their types. In this thesis study, the digital drawings of our interest are construction drawings, there are various types of construction drawings, so that, only seven types of construction drawings are chosen. The work commenced by preparing a dataset of digital drawings to be trained to reach the highest accuracy possible of classification. The network has been trained with the aid of pre-trained models such as VGG16 and VGG19. Additionally, there is detailed information on the drawings such as text and symbols, therefore, an object detection network has been trained to localize the bounding boxes to be able to extract textual or detailed information.

3 Methodology

Fundamentally, this research focus is to classify digital drawings according to their types, the digital drawings used in this research are construction drawings, which consist of considerable types but the main focus of this research is seven types of construction drawings. In this research, there are two neural networks implemented, the first is the neural network for image classification, which is the main classifier that sorts out the images based on their types, while the second one is an object detection network that detects the text from the title boxes of the construction drawings to ensure the consistency of image classification. Before implementation of the neural networks, there are some procedures taken first to be able to train the neural networks as shown in Figure 3.1, which illustrates the sequence of the project and how the models are chosen.

In the following section, it is discussed in detail the source of the datasets collected and a detailed description of how they are labeled and prepared to fit each neural network.

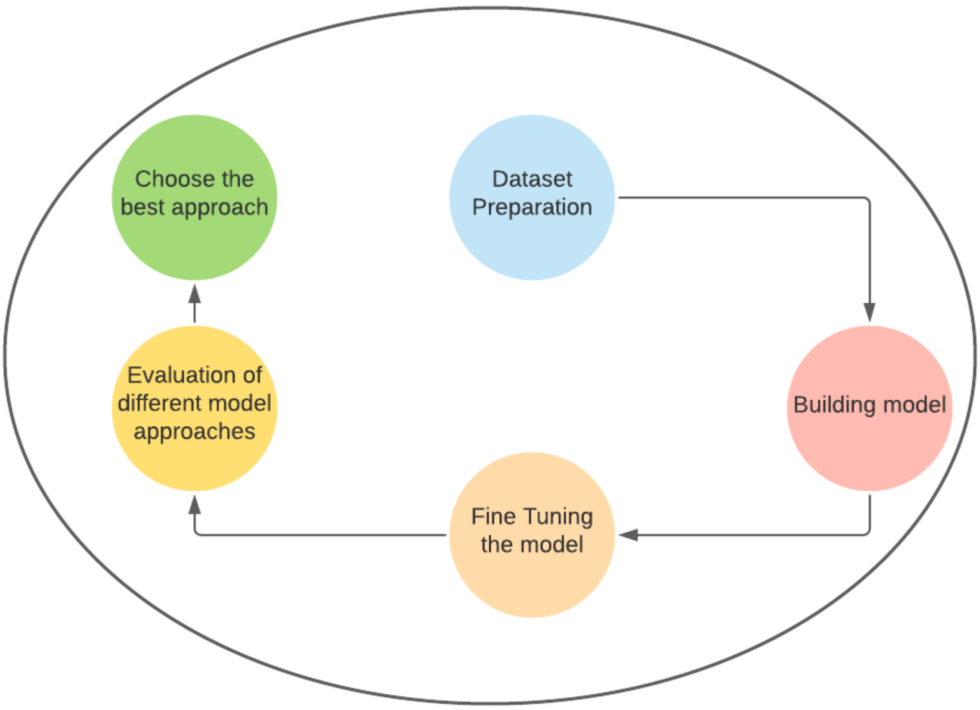


Figure 3.1 Workflow of neural network (Generated Figure)

3.1 Aim and Objectives

This study aims to utilize a deep learning algorithm for feature extraction in construction drawings classification and to achieve the best classification performance by verification of drawings using an object detection approach.

The objectives of the study:

- Build a neural network for image classification of construction drawings.
- Build a neural network for object detection of construction drawings.
- Compare different approaches of CNN architectures for image classification.
- Compare the influence of different alternatives in the object detection network.
- Determine the best approach regarding performance for image classification and object detection networks.

The study has conducted trials using different approaches to achieve the highest performance. Regarding image classification, the neural network has been trained on two CNN architectures, which are VGG16 and VGG19 and each one of them has different variables of data augmentation that has been adapted as well seeking for the best performance. On the other side, object recognition has been trained on two pre-trained networks, which are yolov5s and yolov5m, hence, adapting the variables for both. Before training neural networks, the dataset for both neural networks have been prepared, thus, in the next sections every detail is discussed of how the dataset is collected and labeled; furthermore, the pre-trained CNNs used and their different cases that have been implemented to attain the highest accuracy.

3.2 Datasets for Evaluation

3.2.1 Image classification Dataset

Foremost, the preparation of the dataset is the initialization step to proceed with the deep neural network. The objective is to collect as many construction drawings as possible but due to the various types of construction drawings as mentioned in section 2.2, this study limits the types used to seven types of construction drawings. Following, the types chosen to be implemented in our dataset were Column layout, Detail drawing, Electrical drawing, Elevation plan, Floor plan, Plumbing, and Section drawings. The struggle was in collecting these drawings cause of the rare availability of these types of drawings, therefore, the dataset was collected from different sources.

Accordingly, there were two sources of the dataset, first one was a collection of all available images for the drawings required using google batch on the web to gather as many drawings as possible on the internet. However, this method didn’t provide appropriate drawings as the size of the images varied, hence, some images needed to be downloaded once again to ensure the balance of the dataset and the symmetry of the images.

Subsequently, the other source was by the help of Jimmy Abualdenien by providing Revit models, which are completely done, hence, images can be extracted from these models. The Revit models helped in supporting some types of construction drawings but other types couldn’t be extracted from them due to the unavailability of these drawings in models, for instance, the drawings that have been extracted from the models were Floor plans, Elevations, Section drawings, and Column layouts, while the other types were extracted only using the former source.

The collection of the dataset is a significant step to permit the deep neural network to achieve the highest possible performance, therefore, the neatness and the equality of images sizes was a prerequisite during dataset collecting as any misbalance or in-accurate images will ruin the training of the network. In the following figures, the seven types of construction drawings are illustrated with some samples from the dataset.

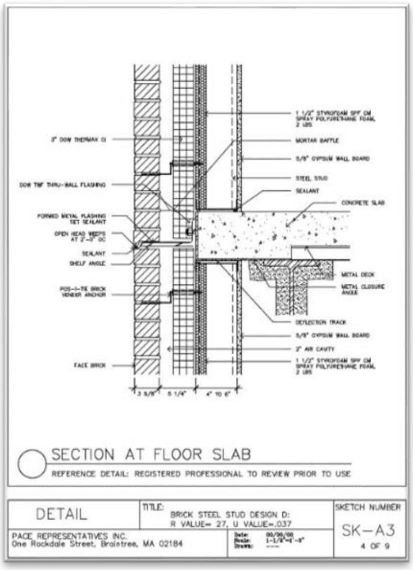


Figure 3.2 An example of Detail drawing (Dataset)

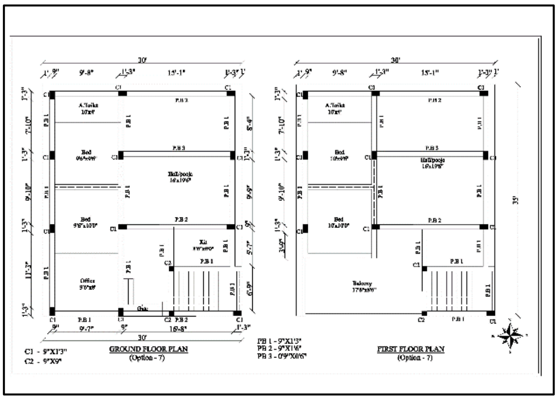


Figure 3.3 An example of Column layout (Dataset)

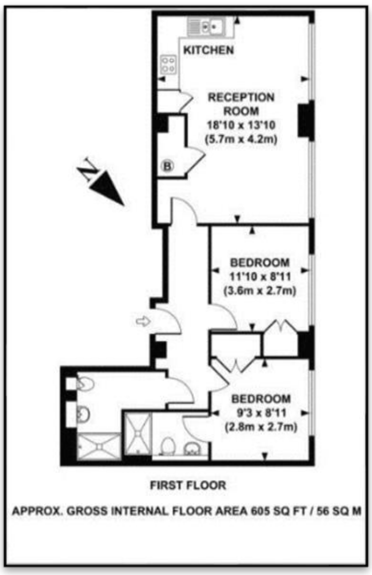


Figure 3.4 An example of a Floor plan (Dataset)

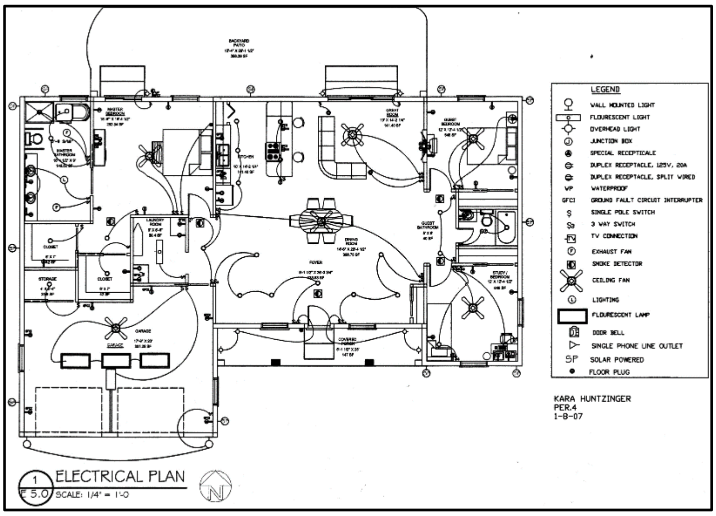


Figure 3.5 An example of an Electrical plan (Dataset)

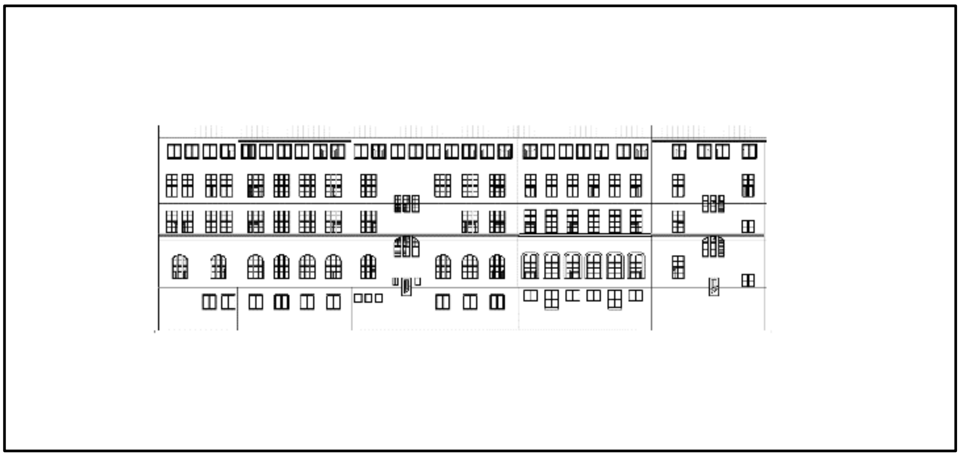


Figure 3.6 An example of an Elevation plan (Dataset)

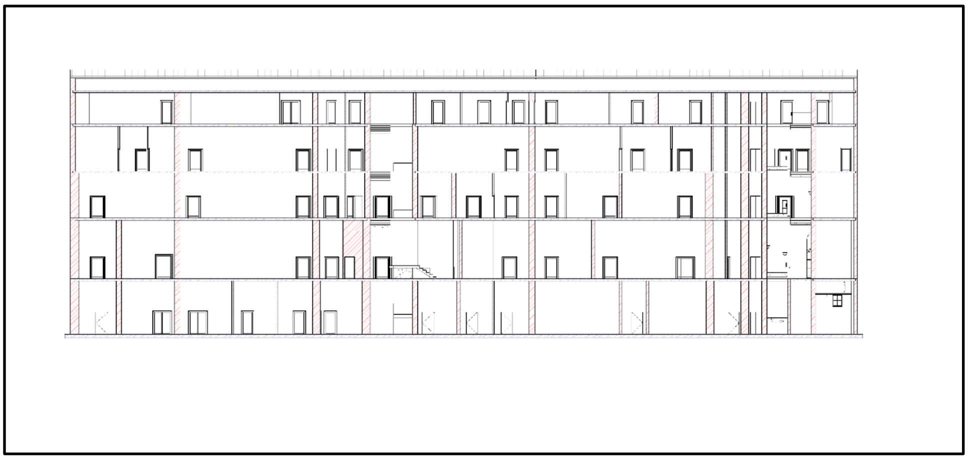


Figure 3.7 An example of Section drawing (Dataset)

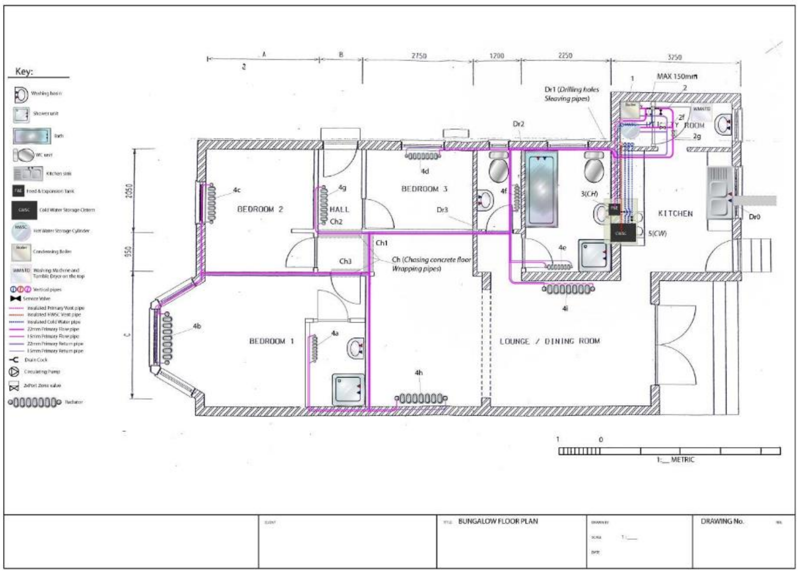


Figure 3.8 An example of a Plumbing plan (Dataset)

3.2.2 Object recognition Dataset

The dataset of object detection has some features that differ from the image classification dataset. The prerequisite of this dataset is that each drawing has a title box that defines the drawing from its type, company, etc. Hence, only a few images in the image classification dataset have these features, thus, the solution is to form our dataset.

The sources of the dataset were almost the same as image classification but the object detection network was run on only three types of construction drawings, which are Floor plan, Elevation, and Section drawings. Consequently, the images are collected from the two sources and the next step is to make sure that all these images have a title box describing the drawing. The images collected are implemented in Autodesk Revit to draw title boxes for all these drawings putting in mind to create different orientations of the title box and different styles to make a variation that would be beneficial in achieving high performance when running the neural network. Figure 3.10 shows a sample of the customized drawings with title boxes.

Following, the images are now ready to be annotated, the annotation step is mandatory in object detection as images are labeled with bounding boxes, therefore, the neural network has a reference of labeled images and then the neural network can predict images after a while. As shown in Figure 3.9, this is how the images are labeled, thereby, every image has label data to be implemented in the neural network.

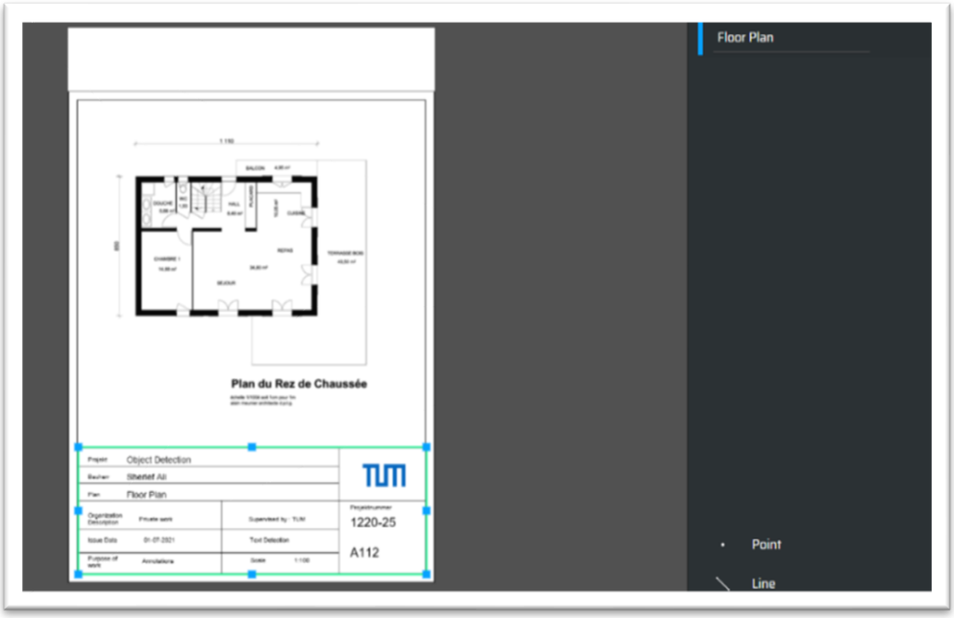


Figure 3.9 Annotating Images of the dataset (Generated image)

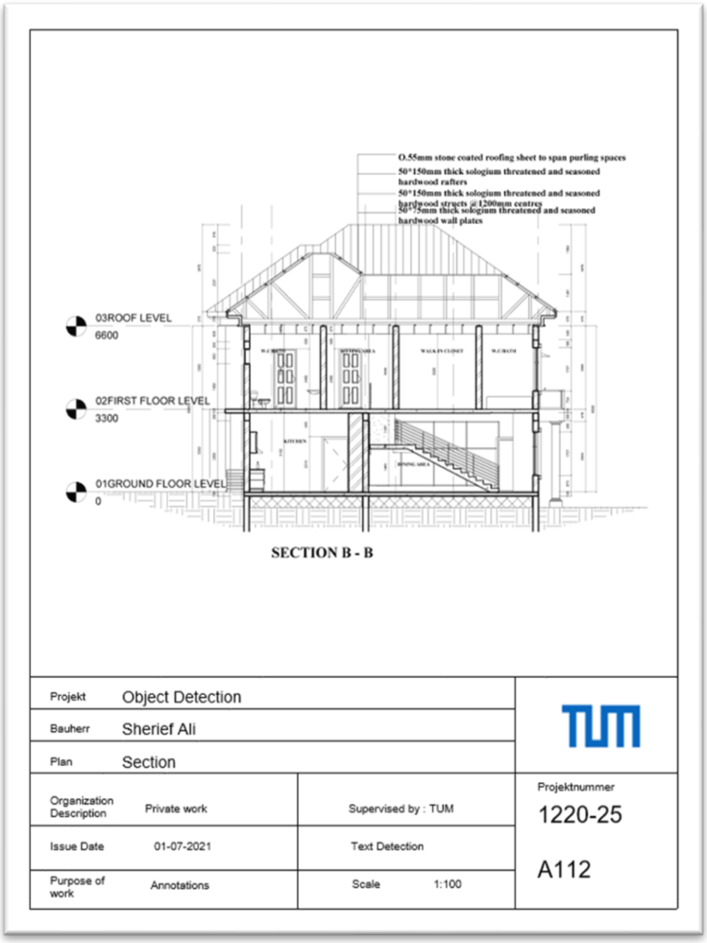


Figure 3.10 Sample of Customized title box for Section drawing (Generated Image)

4 Image classification Network

4.1 Neural Network Structure

The deep neural network is structured to feed in inputs that are trained by extracting features then giving outputs. Headmost, random weights are set in the model together with the provision of inputs to the network. The network training has two folds target labels and model outputs so that the difference between them defines an error. The determination of the effect of each neuron to cause an error is possible using backpropagation, subsequently, the weights are altered. Therefore, this process is repeated several times until reaching the desired goal of the least errors possible. There are diverse parameters that influence the training accuracy and how the network is learning by the time, these parameters are called hyperparameters, for instance, the epochs, which define the number of iterations the network will operate, additionally, the learning rate that interprets the steps taken in each iteration (Nash, Drummond, & Birbilis, 2018).

Forsooth, some architectures are pre-set and configured to train a set of images, accordingly, this study has trained the neural network of image classification using two architectures, which are VGG16 and VGG19 whose structures are discussed in detail in the next sections.

4.2 VGG16 and VGG19 Architecture

Foremost, the model has been trained on VGG16 and VGG19 architectures. VGG16 is composed of 13 convolutional layers and 3 fully connected layers, where the convolutional layers are 3x3 layers having a stride size of 1 with the same padding; furthermore, pooling layers are 2x2 having a stride size of 2. Accordingly, the size of the feature map decreases to half after every single pooling layer and the last feature map is 7x7 having 512 channels. The image sizes of the dataset are set to 224x224 as this is the default of VGG16 architecture. VGG16 architecture is set to train 1000 classes but in this study, seven types of construction drawings are discussed, thus, having only seven classes, therefore, the input of the last layer of the architecture has been changed from 1000 classes to 7 classes. As shown in Table 4.1, this is the final structure of VGG16 architecture which has been adapted according to our model.

In the same manner, VGG19 architecture is almost the same as VGG16 but the only difference is that VGG19 has three more convolutional 3x3 layers in the last three blocks. Figure 4.1 and Figure 4.2 illustrate the difference between VGG16 and VGG19 (Guan et al., 2019).

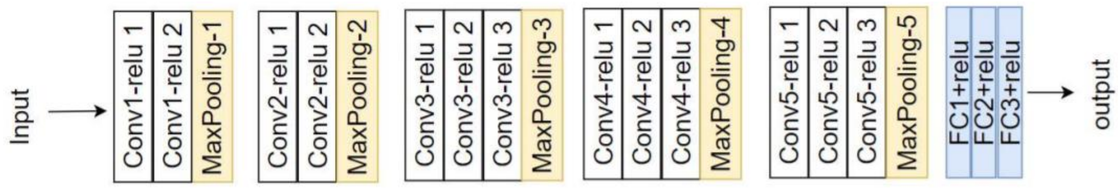


Figure 4.1 VGG16 architecture (Deng, Goy, Li, Arthur & Barbastathis, 2020)

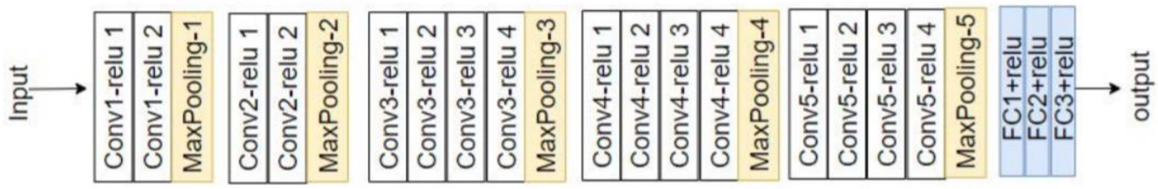


Figure 4.2 VGG19 architecture (Deng, Goy, Li, Arthur & Barbastathis, 2020)

Layer	Patch size	Input size
conv×2	3×3/1	3×224×224
pool	2×2	64×224×224
conv×2	3×3/1	64×112×112
pool	2×2	128×112×112
conv×3	3×3/1	128×56×56
pool	2×2	256×56×56
conv×3	3×3/1	256×28×28
pool	2×2	512×28×28
conv×3	3×3/1	512×14×14
pool	2×2	512×14×14
fc	25088×4096	25088
fc	4096×4096	4096
fc	4096×2	4096
softmax	classifier	7

Table 4.1 VGG16 architecture used in the model (Guan et al., 2019)

5 image classification Approaches

As mentioned before image classification network has been trained on two architectures VGG19 and VGG16. To reach the highest accuracy possible of image classification, the model has been trained using different optimizers and diverse data augmentation combinations. The model has been trained on various learning rate values, different batch sizes, dropout layers, and adjusting regularizer layers as well. The image sizes of the images are set to 224x224. Therefore, all these variables are discussed in detail in the following cases accordingly by defining the parameters and the variables used.

5.1 VGG19 architecture Case 1

The objective of case 1 is to run the model on VGG19 architecture to comprehend the performance of the model on this architecture. The optimizer chosen for this case is RMSprop which has a learning rate of 0.01. The model is trained in this case using a batch size of 16 as being aware of the size of the dataset used. There are no dropout layers or regularizers layers are implemented, to get first an overview of the issues that might arise and then these layers could be implemented in the following cases. The network has an early stopping callback to prevent the model from overfitting or underfitting, ergo the model will stop further training if the model isn't learning anymore, additionally, the patience margin of early stopping is 10, which defines that the model will stop training if the model has stopped improving within 10 epochs. The data augmentation for training is as follows rescale=1/255, shearing range= 0.3, zoom range = 0.3, and horizontal flip as well.

5.2 VGG19 architecture Case 2

The concern of this case is to fine-tune the neural network in conjunction with carrying out different constraints in training the model. The optimizer used in this case has been changed from RMSprop to Adam to compare the influence of both optimizers on classification accuracy. Additionally, the new learning rate is 0.001 to check whether it has a good or bad influence on network learning. In case 1, the model has been trained without dropout layers or batch normalization layers, on the contrary, in this case, the model has two dropout layers of a value 0.5, the first one is under the first fully connected layer and the second one is after the second fully connected layer. The batch size of this case is 8 to investigate the differences between the batch size of 8 and 16. The data augmentation for training is as follows rescale=1/255, shearing range= 0.3, zoom range = 0.3, and horizontal flip as well. The network has an early stopping callback to prevent the model from overfitting or underfitting

5.3 VGG19 architecture Case 3

This case is carried out to train the network on all the possible variables to achieve the best combination for an image classification network. The network has been trained using Adam optimizer with no batch normalization layers implemented. The network has one dropout layer of 0.5 that has been implemented after the second fully connected layer. The learning rate has been furtherly decreased to 0.0001. Regarding the batch size, it has been kept the same as case 2 of a value 8. The data augmentation for training is as follows rotation range=15, rescale=1/255, shearing range= 0.1, zoom range = 0.2, and horizontal flip as well. The network has an early stopping callback to prevent the model from overfitting or underfitting.

5.4 VGG16 architecture Case 1

The reachability of the best performance is the objective of the carried-out cases. As the network has been trained previously on the VGG19 network considering different variables and constraints, therefore, the architecture of the network has been changed to VGG16 architecture running through all different constraints possible. This case has been trained on VGG16 architecture using an RMSprop optimizer. The network has been trained without adding any dropout layers and no regularizers but two layers of batch normalization are added. The first layer of batch normalization is added after the first fully connected layer and the second layer is added after the second fully connected layer. The learning rate value used in this case is 0.01, while the batch size is 8. Regarding the data augmentation for training, it is as follows rotation range= 15, rescale=1/255, shearing range= 0.1, zoom range = 0.1, and horizontal flip as well. The network has an early stopping callback to prevent the model from overfitting or underfitting.

5.5 VGG16 architecture Case 2

The purpose of this case is to train the network on the same architecture VGG16 but with different parameters and different combinations of data augmentation. In this case, the network has been trained using the optimizer Adam and the network has one dropout layer of 0.5, which is added after the second fully connected layer. Furthermore, two regularizers were added to the first two fully connected layers having the value of 0.0001. The learning rate used in this network has a value of 0.0001 and regarding the batch size, it has been increased from 8 to 16 to cover all the instances and achieve the fittest model. Regarding the data augmentation for training, the rescale=1/255, shearing range= 0.3, zoom range = 0.3, and horizontal flip as well. The network has an early stopping callback to prevent the model from overfitting or underfitting.

5.6 VGG16 architecture Case 3

More cases are needed to cover all the combinations possible. This case targets to have some variations in the network but the optimizer used is kept the same as Adam optimizer is used. There are no regularizers layers are implemented in the network, while there are two dropout layers and are implemented with a value of 0.5, the first one is after the first fully connected layer and the second one is after the second fully connected layer. Moreover, two batch normalization layers are implemented in the network in the same place as dropout layers. The learning rate of this case has increased to 0.01, on the other hand, the batch size has decreased to 8. Regarding the data augmentation for training, the rescale=1/255, shearing range= 0.3, zoom range = 0.3, and horizontal flip as well. The network has an early stopping callback to prevent the model from overfitting or underfitting.

5.7 VGG16 architecture Case 4

The aim of this case is to furtherly test all the combinations possible for the parameters. The optimizer used for this case is RMSprop and adding one dropout layer of a value 0.5 after the second fully connected layer and a regularizes is added for the first two fully connected layers as the regularizes have the value of 0.0001. On the contrary, no batch normalization layers are added to this network and regarding the learning rate, its value has been decreased to 0.0001. The batch size used in this case is 16. Regarding the data augmentation for training, the rescale=1/255, shearing range= 0.3, zoom range = 0.3, and horizontal flip as well. The network has an early stopping callback to prevent the model from overfitting or underfitting.

6 Object Detection Network

Considering the object detection neural network, the dataset of the object detection has been trained on YOLO (You Only Look Once), YOLO's approach is applying single CNN to inputs, which convert images into a grid (Shinde, Kothari, & Gupta, 2018). As discussed in section 2.7, the dataset is annotated by labeling bounding boxes as the dataset is composed of a partition for training and another for the validation, where each one of them has a folder that has labels data of the images. The size of the input image is adapted according to the model, while YOLO default size of images is 448*448. Subsequently, the images are trained in the convolutional neural network providing an output of 7x7x30 tensor, which outlines the coordinate of the bounding boxes and the probability distribution (Shinde, Kothari, & Gupta, 2018). Accordingly, the training of YOLO has two outputs, the first output is the labeled images according to the inputs, while the other output is the predicted labels according to the training of the network. Thus, YOLO architecture is illustrated in Figure 6.1.

Certainly, YOLO has pre-trained models that can detect more features from a dataset. In this study, two types of pretrained models are used, which are YOLOv5m and YOLO5s. These two pre-trained models have minor differences as both of them have the default input size of images 640x640 pixels, while YOLOv5m is bigger cause it has more parameters of 21.4 and YOLOv5s has 7.3 parameters, therefore, occasionally in most cases, YOLOv5m has performed better than YOLOv5s.

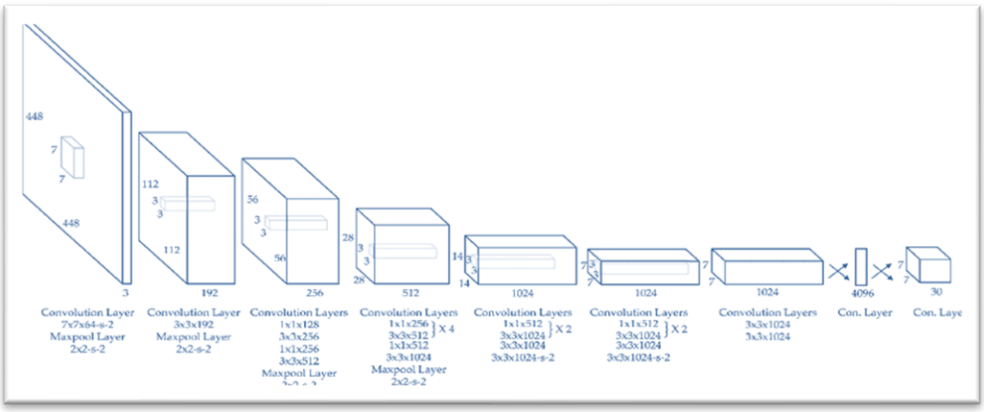


Figure 6.1 YOLO architecture (Koylo, Zhao, & Shao, 2019)

7 Object Detection Approaches

7.1 Elementary object detection YOLOv5s (Trial 1)

Regarding the object detection approach, this approach focused on knowing the performance of the neural network with our dataset and getting familiar with the errors available in the dataset or in the network itself. Therefore, the neural network has been trained on the original parameters of YOLOv5s with a moderate number of iterations as the network has been trained on 200 epochs with a batch size of 8 and the image size is 640.

7.2 Elementary object detection YOLOv5s (Trial 2)

The second trial focused on tuning hyperparameters of the network to make the network able to detect the bounding boxes of the inputs and predict labels. Accordingly, the network can learn but it needs more iteration, hence, the number of epochs has increased from 200 epochs to 1000 epochs. Additionally, the image size input of trial 2 has decreased from 640 pixels to 416 pixels as (Luke & Joseph, 2019) has concluded that the image sizes alter the performance of the network. However, the batch size of the network has been kept the same, which is 8 is due to the small size of the dataset.

7.3 Elementary object detection YOLOv5s (Trial 3)

The concern of this trial is to try most of the cases of tuning hyperparameters to reach the highest possible results from the pre-trained model YOLO5vs. The hyperparameter changed in this trial is the batch size as it has been changed from 8 to 16, while the other parameters are kept the same as the number of iterations is 1000 epochs and the image size is 416 pixels.

7.4 Advanced object detection YOLOv5m (Trial 1)

This approach targeted training the network on another pre-trained model (YOLOv5m) that is discussed before in section 2.7, the concern is to train the network based on the best combination got from the first 3 trials. Therefore, the network has been trained for 1000 epochs with image sizes 416 pixels. The batch size is set in this trial to 8 as a first attempt to check out the results from this pretrained model.

7.5 Advanced object detection YOLOv5m (Trial 2)

This trial aims to reach the highest accuracy possible by running different cases and using different approaches. Consequently, the network has been trained using almost the same parameters as in trial 1 but the batch size only has been changed from 8 to 16 to cover all the cases possible for the pretrained model YOLOv5m, while epochs are 1000 as in trial 1 and image size is 416 pixels as well.

8 Results and Discussion

8.1 Metrics for Performance Measurement

Deep learning models are evaluated using diverse methods, the methods used to evaluate our model are confusion matrix, accuracy, precision, recall, and F1-score. These methods are used to judge the power of classification of the model. Additionally, the display of the confusion matrix provides an overview of how the model performed in the classification of images (Flach, 2019).

The data derived from the model is divided into two classes in binary classification, the two classes are positives (P) and negatives (N) as in Figure 8.1. Accordingly, each class has two predictions, hence, the binary classifier has four outcomes, which are true positives (TP), true negatives (TN), false positive (FP), and false-negative (FN) as in Figure 8.2. True positive and true negative are correct classification of the model as true positive is a correct prediction of positive class, while true negative is a correct prediction of negative class. Similarly, false positive is an incorrect prediction of positive class and false negative is an incorrect prediction of negative class (Saito & Rehmsmeier, 2015).

In the next sections, all the evaluation methods are discussed in detail to give an overview of how our model is evaluated.

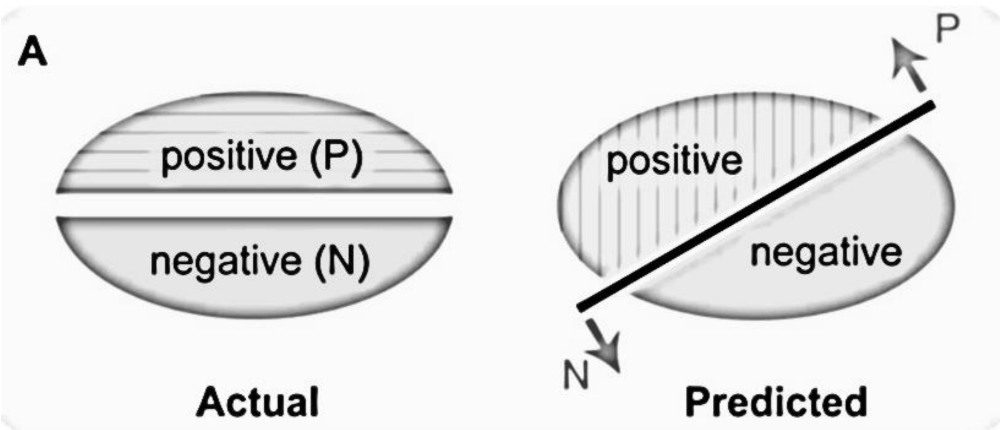


Figure 8.1 Classes of binary classification (Saito & Rehmsmeier, 2015)

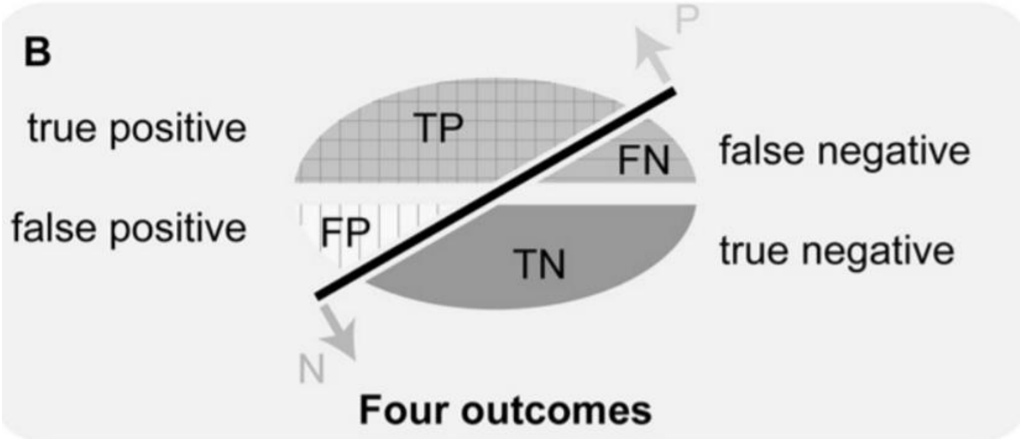


Figure 8.2 outcomes of binary classification (Saito & Rehmsmeier, 2015)

The outcomes of binary classification determine whether the dataset is balanced or not and reveals if the images are classified correctly. Figure 8.3 shows a sample of values of balanced and imbalanced classification.

Figure 8.3 Example of balanced and imbalanced classification (Saito & Rehmsmeier, 2015)

8.1.1 Classification Report

The classification report has different parameters, which define the precision of classification. First of all, the precision value, which is the percentage of positive instances out of total predicted positive, while the recall is the percentage of positive instances out of the total actual positive instances and regarding the F1 score, it is the mean of precision and recalls considering both as the higher F1 score, the better classification performance, while the accuracy is a metric value, which it is a simple illustration of the performance of the model (Flach, 2019).

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{8.1}$$

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{8.2}$$

$$F1\ score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2 * \text{Precision} * \text{Recall}}{Precision + Recall} \tag{8.3}$$

$$Accuracy = \frac{\text{TP} + \text{TN}}{TP + FP + TN + FN} \tag{8.4}$$

8.1.2 Confusion Matrix Visualization

The visualization of a confusion matrix is a principal part of the evaluation of the deep learning model as the outcomes of binary classifier (TP, TN, FP, FN) are shown on a diagram to reveal misclassed images, hence, the classification can be improved to reach the highest accuracy of image classification. As shown in Figure 8.4, the true positive and other metrics are demonstrated on the diagram where the y-axis is the predicted classes and the x-axis defines the actual classes. Therefore, the intersection between the predicted and actual classes gives true positive values.

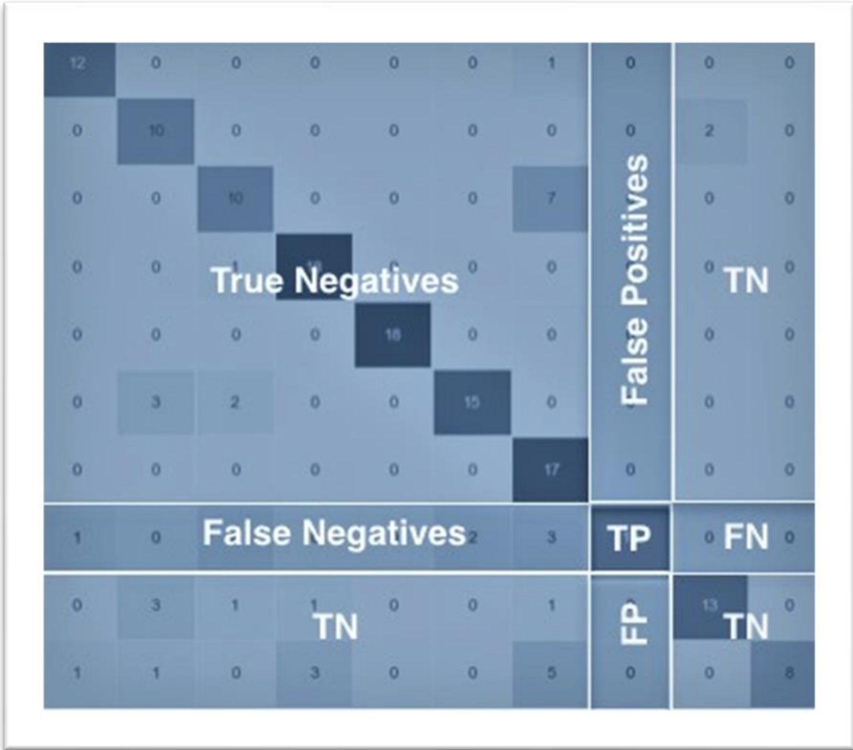


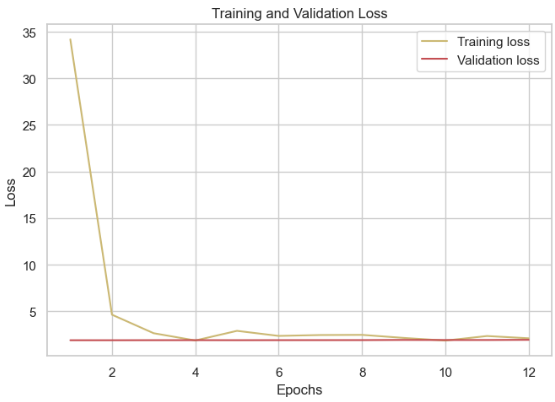
Figure 8.4 Elaboration of Confusion matrix display (Loukas, 2020)

8.2 Image Classification Approaches

8.2.1 VGG19 architecture Case 1

The concern of case 1 is to run the model on VGG19 architecture without considering varying data augmentation or normalization of the curve as it is just a trial to check out how the model performs with the dataset.

The model performance according to loss and accuracy is illustrated in Figure 8.5. The model performance was not good, as the validation loss curve stayed the same the whole training process, while the training loss curve has begun from a high value and then had a sudden decrease in values, which is an indication of misclassification of data. On the other hand, the training and validation accuracy curves show lots of sudden ups and downs revealing instability in the model. Furthermore, the model has stopped training after 12 epochs to prevent overfitting as the model has an early stop function, which stopped the model as it isn't improving anymore.



(1)

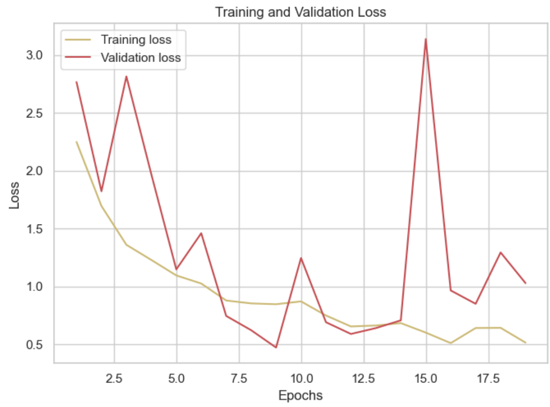
(2)

Figure 8.5 (1) Training and validation loss curve, (2) training and validation accuracy curve Case 1

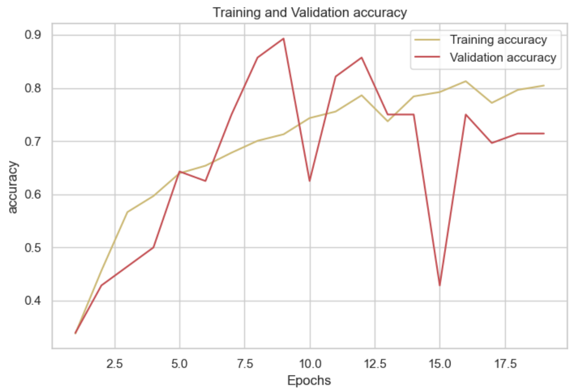
8.2.2 VGG19 architecture Case 2

The focus of this case is to redress the wrongs found in case 1. The usage of another optimizer can be a significant adjustment to network training; moreover, the batch normalization layers can reduce the generalization of the training, hence, achieving better accuracy together with the added dropout layers that are added to overcome the overfitting that has happened in the first case.

Accordingly, Figure 8.6 shows the training and validation loss curves change from case 1, the training curve shows a stepping down in values not like in the first case, which has shown a sudden decrease in values. While the validation loss curve shows a decrease of loss but having some peaks of increase in values, which defines the inaccurate classification of images. Regarding training and validation accuracy curves, the curves show a clear performance of overfitting as the validation accuracy curve has exceeded the training accuracy curve in many points then the model has stopped training due to no further improvement of the model.



(1)



(2)

Figure 8.6 Training and validation loss curve, (2) training and validation accuracy curve Case 2

8.2.3 VGG19 architecture Case 3

The intention of case 3 is to carry out all the variations possible regarding data augmentation and normalization of curves in order to decide whether VGG19 architecture is possible to classify the dataset or the architecture can't find a solution to accurately classify the types of drawings. Therefore, the changes in the model were in the dropout layers by using only one layer of dropout and adding a regularizer to prevent overfitting that happened in the previous two cases and adjusting the learning rate as well.

The curves in Figure 8.7 have revealed that the model performance has increased and the overfitting issue has been barely overcome as the training and validation loss curves show a gradual decrease in losses reaching 0.125 in the training loss curve and 0.6 in the validation loss curve. While the training accuracy curve has reached a maximum value of 0.99, on the other hand, the validation accuracy curve has reached a maximum of 0.88.

(1)

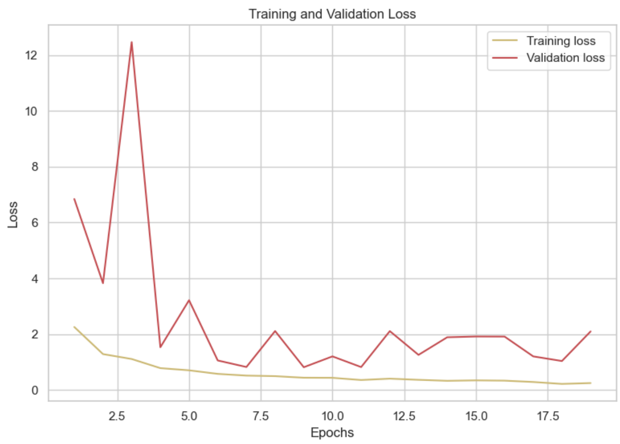
(2)

Figure 8.7 Training and validation loss curve, (2) training and validation accuracy curve Case 3

8.2.4 VGG16 architecture Case 1

In the previous three cases, the model has been trained on VGG19 architecture running through all the possible parameters, for this reason, the architecture of the network has been changed to VGG16. The data augmentation of the network has been varied accordingly and proceeding this case with RMSprop optimizer.

The performance of the network is far away from the accurate classification of images as Figure 8.8 shows the training and validation loss curves not approaching and training loss curve reached the minimum by 0.12, while validation loss reached 1. The curves show that there is no overfitting has happened in the training as the training and validation accuracy curve are proceeding appropriately, but the model has stopped training because the model has stopped improving.



(1)

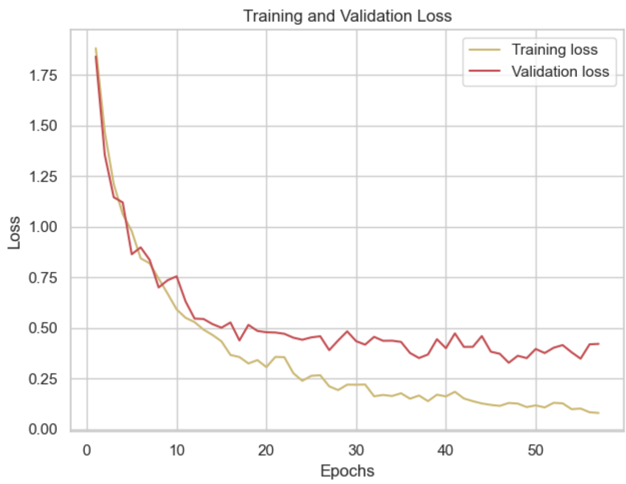
(2)

Figure 8.8 Training and validation loss curve, (2) training and validation accuracy curve Case 1

8.2.5 VGG16 architecture Case 2

Some adjustments have been made to the model in this case as the optimizer has been changed to check the performance of the network to various optimizers plus adding one dropout layer. Two regularizers are added to fully connected layers and the learning rate has been reduced to find out the best value of learning rate for the network.

To illustrate the graphs from case 2, Figure 8.9 shows the validation and training loss curves that displays a good performance of the model as the losses are decreasing gradually without sudden increase or decrease and the values of training and validation losses are near which indicates a good classification of images as the minimum of training loss has reached 0.08 and the minimum of the validation loss is 0.3. Regarding the training and validation accuracy, both the training and validation curves started from almost the same point and increasing gradually, getting nearer to each other and the training accuracy has reached a maximum at 0.98 and the validation accuracy has reached the maximum at 0.94. Therefore, the accuracies are high enough to get a good model for image classification. Furthermore, the model has continued training for 60 epochs, which is a good indication that the model is learning and hasn't stopped.



(1)

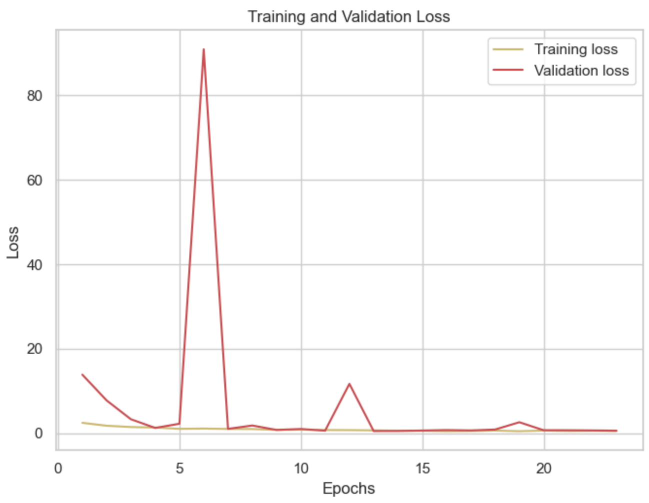
(2)

Figure 8.9 Training and validation loss curve, (2) training and validation accuracy curve Case 2

8.2.6 VGG16 architecture Case 3

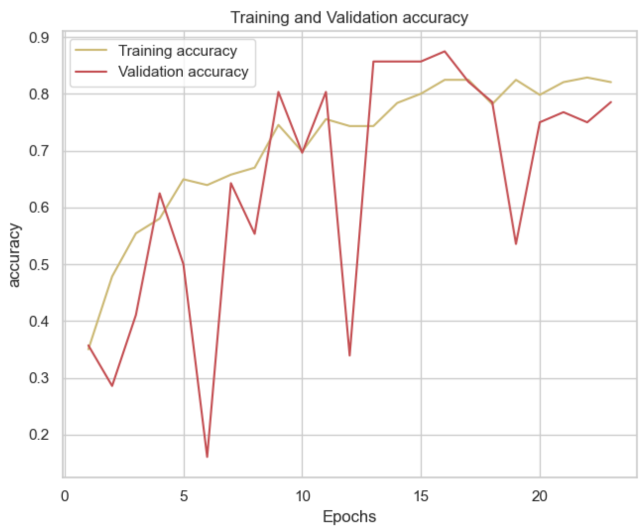
The amendments made in this case were based on the previous case results to test the variables possible to achieve the best performance. This case has a batch normalization layer to reduce the generalization error and the learning rate has been increased to check the best value of learning rate the model performs better and two dropout layers are implemented in the network.

The curves in Figure 8.10 show that the validation loss and the training loss curves are almost on the same line having the same numbers, but the validation loss curve has some peaks, while this doesn’t mean that the model performance is good as it reveals that the network is not learning well. Additionally, the training and validation accuracy curves reveal an overfitting issue in the training as the validation accuracy curve exceeded the training accuracy curve in some points. The training accuracy has reached the maximum at 0.82, while the validation accuracy has reached 0.87.



on

(1)



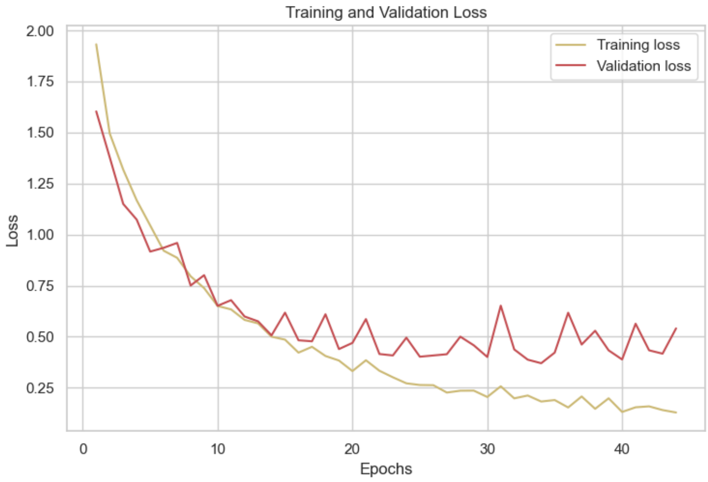
(2)

Figure 8.10 Training and validation loss curve, (2) training and validation accuracy curve Case 3

8.2.7 VGG16 architecture Case 4

Further variations in this case as the optimizer have been changed to ascertain the best combination between the variables and the optimizers. There is only one dropout layer implemented in the model and no batch normalization layers are implemented. There are two regularizers are added to the fully connected layers of the network to prevent overfitting issues and the learning rate has been reduced than in the previous case.

To illustrate the graphs from case 4, the curves in Figure 8.11 reveal the performance of the model, the training and validation loss curves are gradually decreasing and approaching a little bit in some points, which is a good indication of good performance, while the training and validation loss values haven't approached properly as the minimum of training loss is 0.08, while the minimum of the validation loss is 0.42. On the other hand, the training and validation accuracy curves reveal an overfitting issue at the beginning of the training and almost in the middle of training the model as well. The accuracy curve shows instability in the model training as the numbers are decreasing and increasing with high differences, thus, it can be predicted that the model hasn't properly classified the images in addition that the model has stopped training early cause the model isn't learning anymore.



(1)

(2)

Figure 8.11 Training and validation loss curve, (2) training and validation accuracy curve Case 4

8.2.8 Evaluation of different Cases

	Optimizer	Regularizer	Batch Normalization	Dropout	Learning Rate	Shear range	Zoom Range	Rotation angle	Accuracy
VGG19 Case 1	RMsprop	-	-	-	0.01	0.3	0.3	-	0.17
VGG19 Case 2	Adam	-	2 layers of batch norm	2 layers of dropout = 0.5	0.001	0.3	0.3	-	0.71
VGG19 Case 3	Adam	2 layers of regulazier l2=0.0001	-	One layer of dropout= 0.5	0.0001	0.1	0.2	15	0.86
VGG16 Case 1	Rmsprop	-	2 layers of batch norm	-	0.01	0.1	0.1	15	0.68
VGG16 Case 2	Adam	2 layers of regulazier l2=0.0001	-	One layer of dropout= 0.5	0.0001	0.3	0.3	-	0.94
VGG16 Case 3	Adam	-	2 layers of batch norm	2 layers of dropout = 0.5	0.01	0.3	0.3	-	0.8
VGG16 Case 4	Rmsprop	2 layers of regulazier l2=0.0001	-	One layer of dropout= 0.5	0.0001	0.3	0.3	-	0.86

Table 8.1 Evaluation of different cases

	Precision	Recall	F1-score
VGG19 Case 1	0.02	0.14	0.04
VGG19 Case 2	0.79	0.76	0.71
VGG19 Case 3	0.84	0.86	0.82
VGG16 Case 1	0.76	0.73	0.66
VGG16 Case 2	0.95	0.90	0.9
VGG16 Case 3	0.78	0.75	0.75
VGG16 Case 4	0.81	0.88	0.81

Table 8.2 Classification report for the cases

8.2.9 Evaluation of the Confusion matrix

The confusion matrix of the best case (VGG16 case 2) has revealed high true positive values for all types of construction drawings. Nevertheless, the matrix has also revealed that a conflict has happened during classifying some types of drawings, for instance, there is a conflict between Elevation and Detail drawing as shown in Figure 8.12, which shows that there is an issue in six drawings during classification. This conflict might be due to some common features in both drawings as detail drawings might show a detailed part of an Elevation plan, therefore, some conflicts can arise. Figure 8.13 and Figure 8.14 elaborate the idea of how there might be some issues in classifying some drawings of these types.

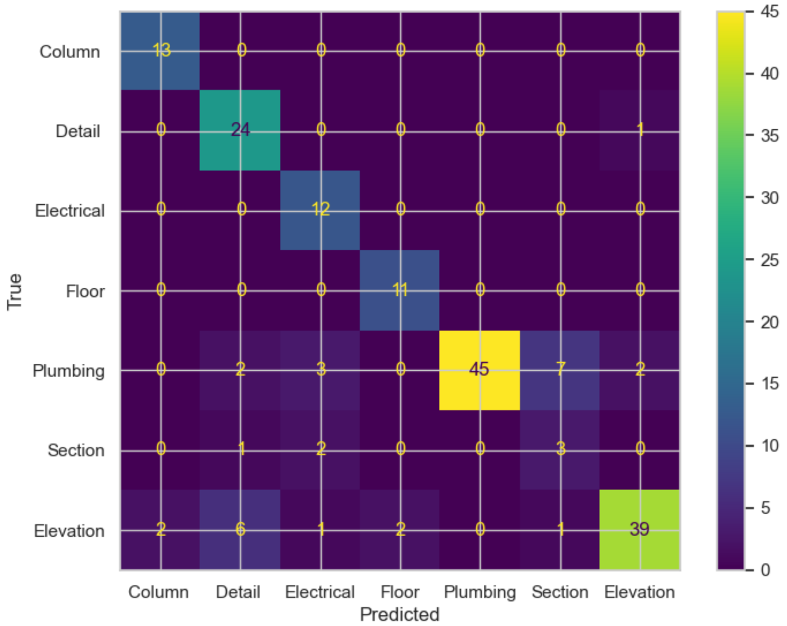


Figure 8.12 Display of Confusion matrix of the best case (Case 2 VGG16)

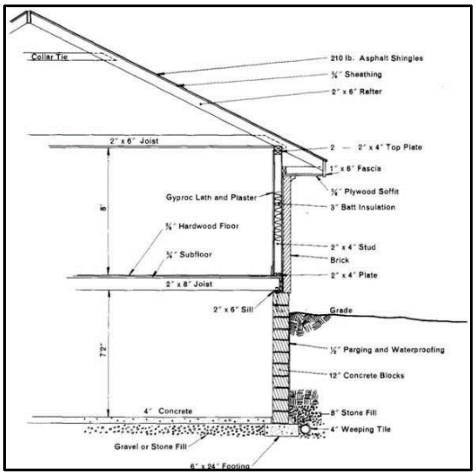


Figure 8.13 Sample of a Detail drawing that has some features from an Elevation plan

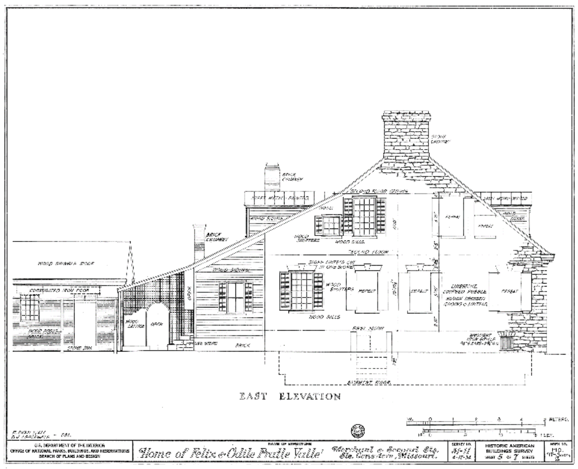


Figure 8.14 Elevation plan that has some features from a Detail drawing

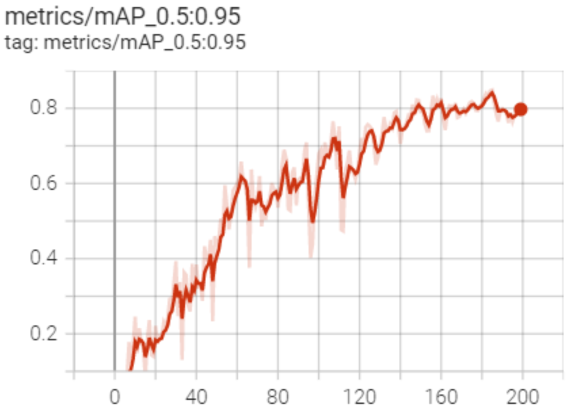
8.3 Object Detection Approaches

The object detection network aimed to localize bounding boxes around the titles boxes of the drawings. The network has been trained to localize bounding boxes and it has achieved an accuracy of 100% as shown in Figure 8.15. Additionally, the network has been trained to localize the bounding boxes and detecting the types of the construction drawing, hence, the results are discussed in the following sections.

Figure 8.15 Localization of the bounding box of the drawing

8.3.1 Elementary object detection YOLOv5s (Trial 1)

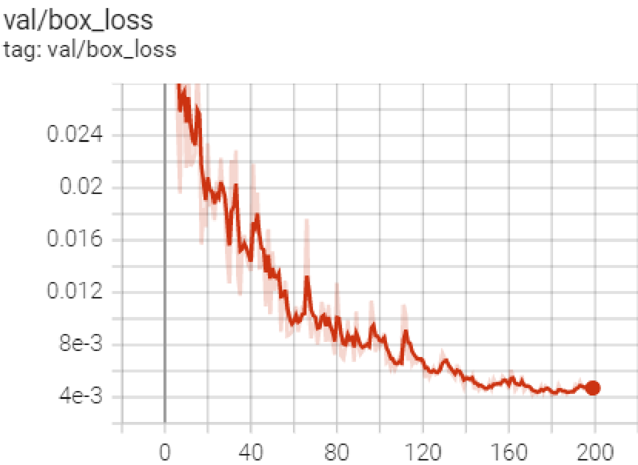
The first trial of training has shown promising results as this trial the mean average precision (mAP) of the model is 82% as illustrated in Figure 8.16 (A) and at the same time the mAP of Floor plans is 94%, mAP of Section drawings is 82%, and mAP of Elevations is 70%. Consequently, the first trial has revealed that the model's performance can be enhanced with some alterations.



(A)

Figure 8.16 (B)(C) shows the training to box loss and validation to box loss and they demonstrate a bit good performance in the first trial as the losses are decreasing gradually by running more iterations of training. As a consequence of training loss approximately reached zero and validation loss reached 0.004.

(B)



(C)

Figure 8.16 Trial 1(A) mean average precision, (B) training loss curve, and (C) Validation loss curve

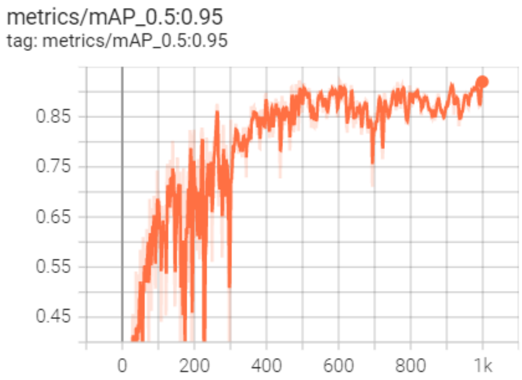
Figure 8.17 is the input image that has its labels annotated, while Figure 8.18 is the image with the predicted label. Hence, these images have shown that the model has incorrectly predicted the labels as the original label of the image is Elevation, while it is predicted as Section drawing. This is due to the low number of iterations and the model needs further alterations to detect the objects correctly of the three types of drawings.

Figure 8.17 Pre-set label for an image

Figure 8.18 Incorrect prediction of the image

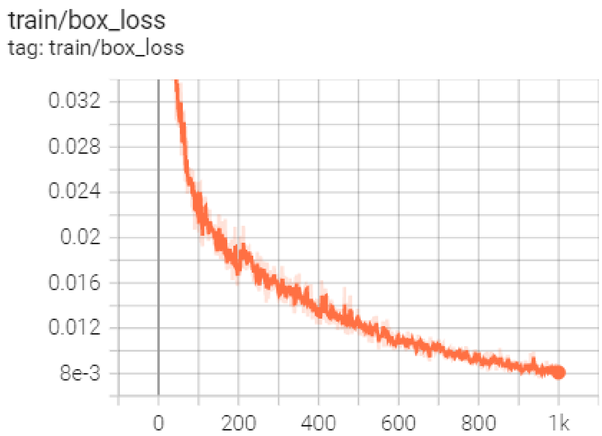
8.3.2 Tuning the hyperparameters of YOLOv5s (Trial 2)

The second trial of training object detection network is to manipulate the training of the first trial and to benefit from the previous results to get more precise results of labels prediction. The second trial has revealed that tuning hyperparameters has a great influence on the model performance, additionally, long iterations of training the model have revealed an increase in model learning. The mAP of Foor plan label detection reached 98% and Section drawings reached 91%, while Elevation plans achieved an accuracy of 89%. Figure 8.19 (A) exhibits the mean average precision of the second trial as the mAP has reached 93%.

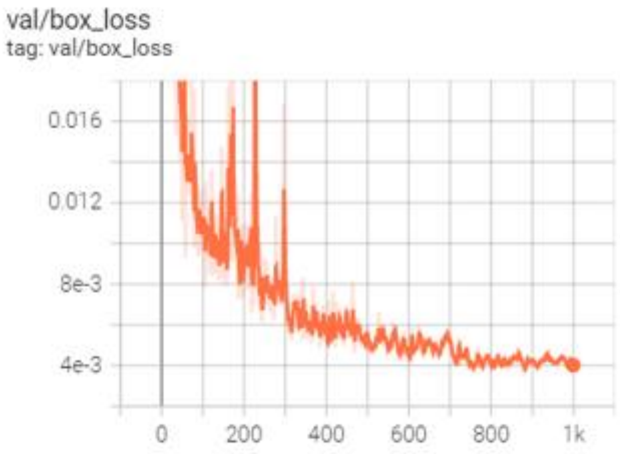


(A)

The training loss and validation loss curves are illustrated in Figure 8.19 (B) (C). The training loss of trial 2 has reached almost zero as in the first trial, while validation loss shows the same final values of reaching 0.004, but having some peaks of increasing loss in the iterations from 200 to 250.



(B)

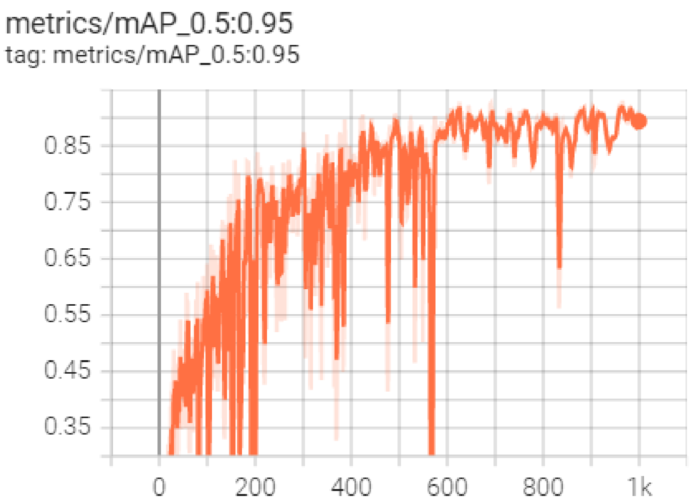


(C)

Figure 8.19 Trial 2 (A) mean average precision, (B) training loss curve, and (C) Validation loss curve

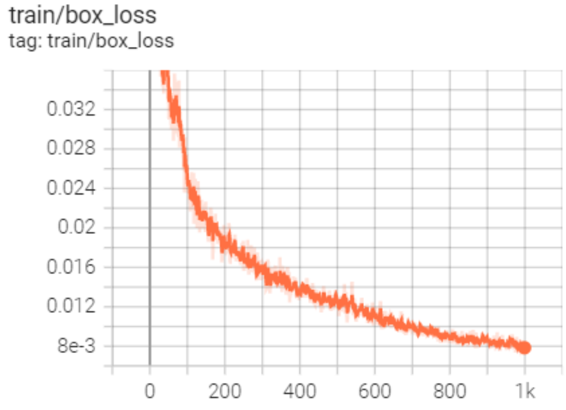
8.3.3 Tuning the hyperparameters of YOLOv5s (Trial 3)

This trial approach is to detect whether YOLOv5s is suitable to train the dataset or it is too small to get valuable results. The hyperparameters didn't change much from the second trial, but to gather all the cases possible for YOLOv5s pretrained model. The results were almost the same as trial 1 and trial 2 or even lower as the mean average precision reached 89%, while the mAP graph has shown a lot of peaks during training as shown in Figure 8.20(A), which detects the instability of the model, therefore, giving inaccurate predictions

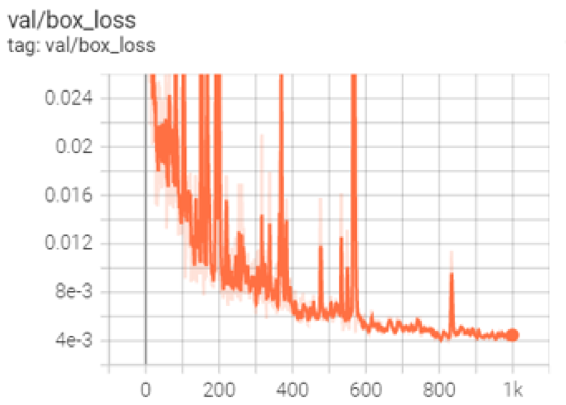


(A)

The training and loss curves in Figure 8.20 (B) (C) reveals almost the same performance as in trials 1 and 2, while validation loss has revealed that it has the highest peaks of loss increase giving a bad sign of correct predictions.



(B)

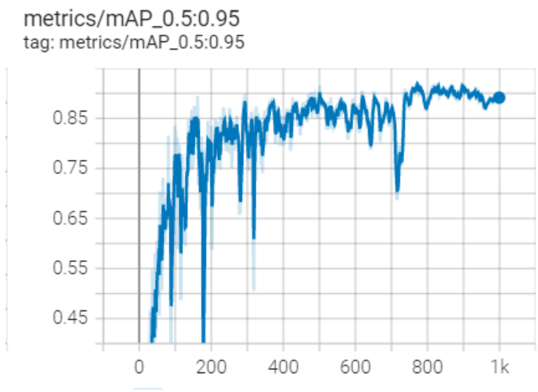


(C)

Figure 8.20 Trial 3 (A) mean average precision, (B) training loss curve, and (C) Validation loss curve

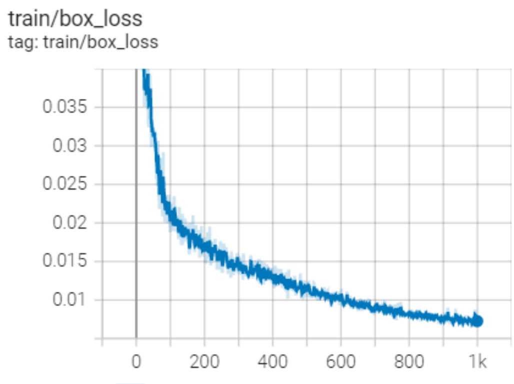
8.3.4 Advanced object detection YOLOv5m (Trial 1)

The procedure taken in this step is to train the model on another pre-trained model, which is YOLOv5m as the results from YOLOv5s have shown instability in validation losses and mean average precision. The first trial of YOLOv5m has revealed a significant increase in the Floor plan's accuracy of reaching 92% and Section drawing's accuracy is 91%,/ while Elevation plans have an accuracy of 82%. Therefore, the results are encouraging that the model may achieve better performance. Figure 8.21 (A) shows the mean average precision of the model, the mAP has reached 89%.

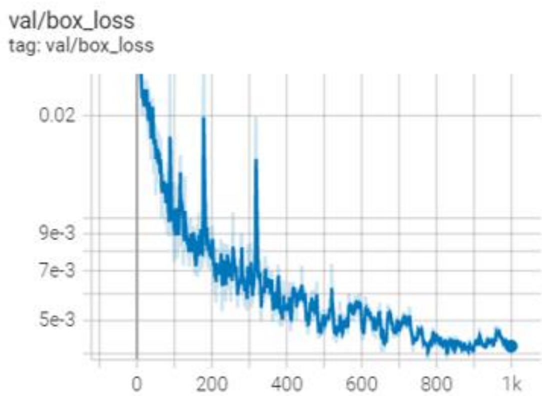


(A)

Regarding the training and validation loss curves, YOLOv5m has ensured stability during training as the training curve is completely stable of reaching 0.005 as shown in Figure 8.21 (B) (C), while validation has some peaks in the beginning, but the curve returned to the normal state again from 250 epochs reaching to almost zero by the end of the curve.



(B)

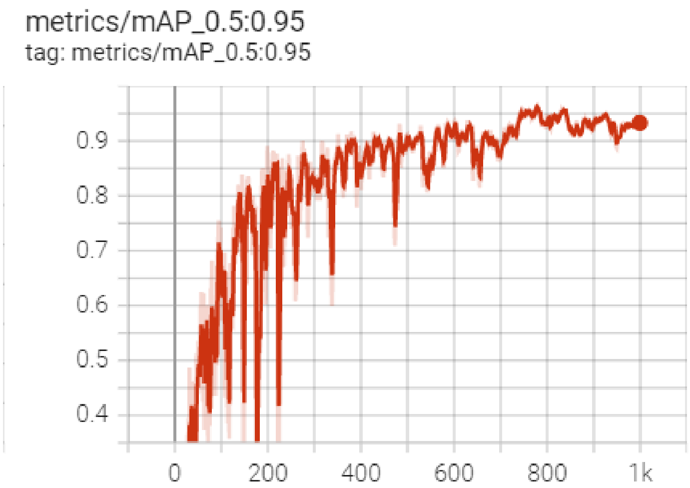


(C)

Figure 8.21 YOLOv5m Trial 1 (A) mean average precision, (B) training loss curve, and (C) Validation loss curve

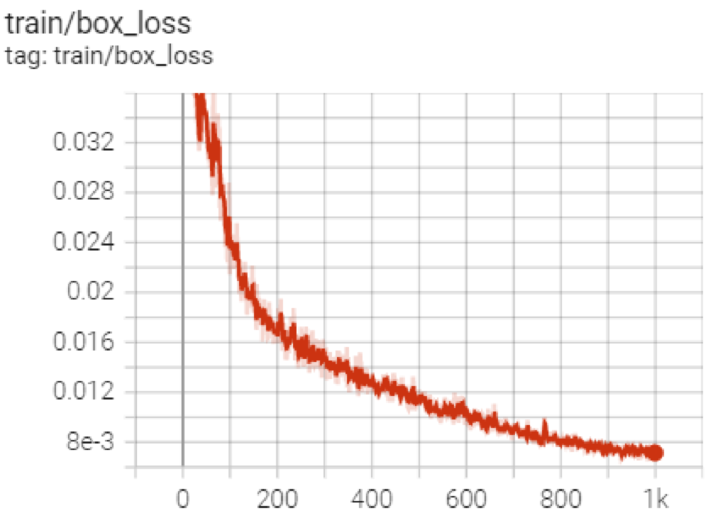
8.3.5 Tuning the hyperparameters of YOLOv5m (Trial 2)

According to the promising results from the first trial of training on YOLOv5m, the hyperparameters are tuned. Therefore, the results have reached the peak of the best performance of the model. The mean average precision of this trial has reached 94% as in Figure 8.22 (A). Furthermore, floor plans accuracy is 98.5%, Section drawings accuracy is 93%, and the Elevations accuracy is 90%.

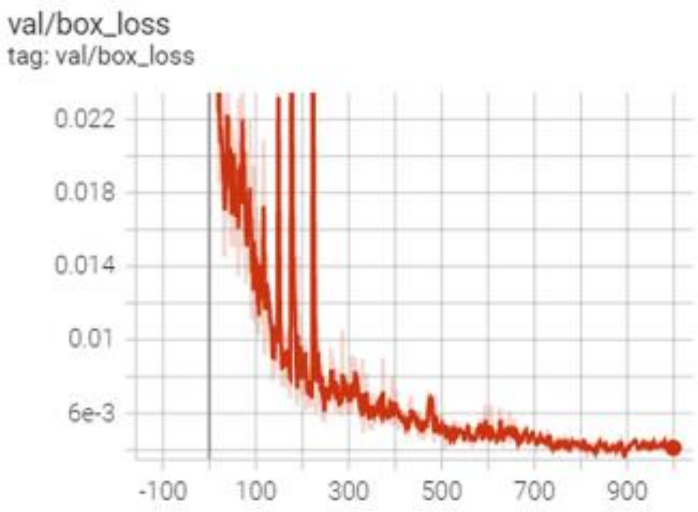


(A)

Concerning the training and validation loss curves, both curves have achieved almost zero percent losses as in Figure 8.22 (B) (C). On the other hand, the validation curve has revealed some peaks only in the beginning but the curve proceeds stably until the end, which proves accurate and precise prediction of labels.



(B)



(C)

Figure 8.22 YOLOv5m Trial 2 (A) mean average precision, (B) training loss curve, and (C) Validation loss curve

Consequently, the trial has appropriately predicted the labels as Figure 8.23 is an image of the pre-set labels and Figure 8.24 is an image showing the predicted labels and as shown the label has been predicted correctly and wasn't misclassed as in section 8.3.1. Accordingly, the label predictions of the three types of drawings have been adequately predicted.

Figure 8.23 Pre-set label for the image

Figure 8.24 Correct label prediction

8.3.6 Evaluation of the trials

Approaches	mAP	Floor Plan mAP	Section mAP	Elevation mAP
Yolov5s Trial 1	82%	94%	82%	70%
Yolov5s Trial 2	93%	98%	91%	89%
Yolov5s Trial 3	89%	94%	91%	80%
Yolov5m Trial 1	89%	92%	91%	82%
Yolov5m Trial 2	**93%**	**98.5%**	**93%**	**90%**

Table 8.3 Evaluation of different trials

8.4 Study Limitations

The purpose of this study is to extract features from digital drawings by training two deep neural networks, the first one aims to classify the images according to their types, while the second network aims to detect objects from the drawings such as textual information. A dataset is collected for running both networks and the struggles were here cause the construction drawings datasets are not available. Thus, the drawings have been collected from different sources as discussed before. Therefore, the dataset collected was limited as the neural networks need large datasets to achieve the highest possible accuracy. Additionally, this study has trained the image classification network on seven types of construction drawings, hence, there were some types of drawings not widely available as much as other types.

9 Conclusion and Future work

9.1 Conclusion

Generally, manual classification of digital drawings is an exhausting task and time-consuming for human beings as some megaprojects have a major number of construction drawings that need to be classified within the shortest timeframe possible. In particular, some projects like a shopping mall or train stations have diverse types of drawings and they have a time limit for image classification. This study proposes a method to automate the image classification process of construction drawings accurately and precisely. Furthermore, this study aims to perform object detection and provides extra information for the end-user by localizing the bounding boxes around the title boxes of the construction drawing. Therefore, the automation of image classification can overcome the consumption of the time that might be taken by classifying the images manually.

Comprehensive approaches are conducted to both types of neural networks to reach the highest accuracy and the best possible results. The image classification neural network has been tested on two CNN architectures, which are VGG19 and VGG16, the neural network's performance with the VGG19 architecture was not very precise and the network has revealed overfitting issues in most cases. The neural network has shown promising performance in early cases of VGG16 until reaching the best case, which has no overfitting and the training and validation accuracy and losses curves have revealed that the model is learning accurately and without sudden peaks or drops in values. Accordingly, the cases are carried out based on fine-tuning of the hyperparameters by adjusting several parameters in each case to find the optimal combination. Different approaches are carried out on object detection neural networks as well, but the object detection network has been tested with two pre-trained models, which are YOLOv5m and YOLOv5s. Consequently, the network has been trained on both of them where each pre-trained model has several trials conducted to adjust the hyperparameters and it has shown precise results of object detection using YOLOv5m.

Overall, this thesis study presents two neural networks for image classification and consistency verification as the image classification network has reached 94% and the object detection network has achieved an accuracy of 93%.

9.2 Future Work

This study has conducted diverse experiments on two neural networks, which are image classification and object detection networks. The image classification network has extracted features from the digital drawings, therefore, the network can recognize and categorize the digital drawings according to their types. While the object detection network has localized the position of bounding boxes through training the network on annotated images, thus, the network has learned and can detect objects from drawings accurately.

Accordingly, image classification, as well as object detection networks, are beneficial in different fields by reducing the consumption of time. With the help of the image classification network, 3D models of construction projects can be obtained from the classified 2D drawings. This is especially interesting for construction projects that do not have 3D models. Figure 9.1 illustrates the usage of classified 2D drawings in forming a 3D model. Whilst the object detection network can be used to localize the information in the digital drawings, hence, the crucial information can be extracted from the title boxes, for instance, the text defining the drawing or the dimensions.

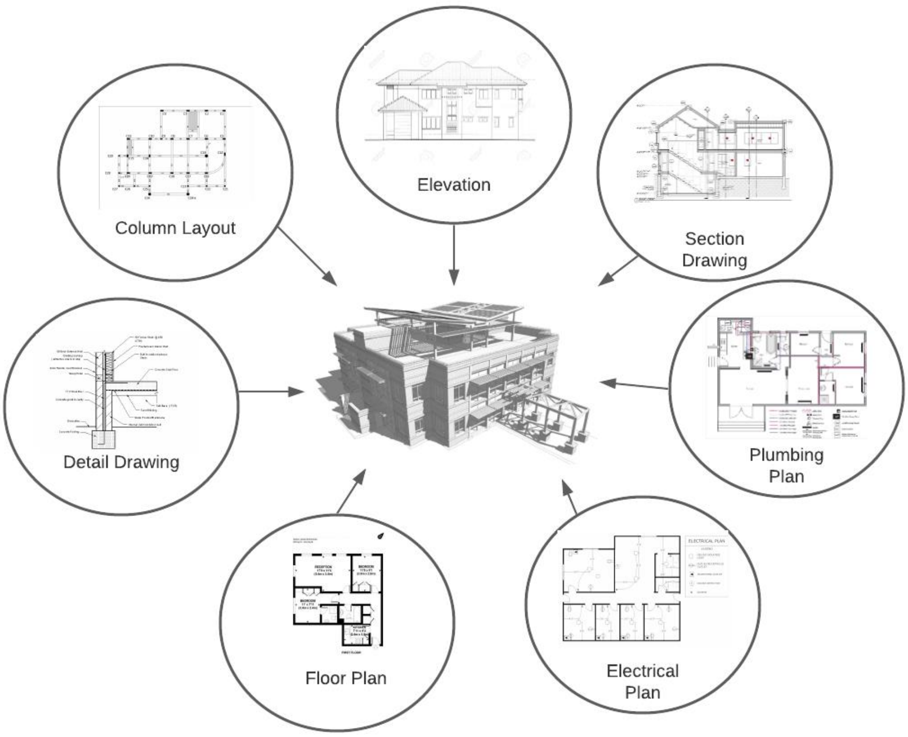


Figure 9.1 Forming a 3D model from 2D digital drawings

Furthermore, (Borrmann, Abualdenien, & Krijnen, 2021) has proposed an approach of linking digital drawings to BIM models, the basic idea is that an action or change in the digital drawings corresponds to an action or change in the BIM model, as shown in Figure 9.2. The usage of both neural networks in this study can be beneficial in this approach as the digital drawings can primarily be classified according to their types and the textual information and dimensions can also be extracted from the digital drawings to provide accurate and precise information as well as categorization of the drawings.

Figure 9.2 Linking 2D drawing to 3D model (Borrmann, Abualdenien, & Krijnen, 2021)

References

Borrmann, A., Abualdenien, J., & Krijnen, T. (2021). Proc. of the CIB W78 Conference 2021. Information Containers Providing Deep Linkage of Drawings and BIM Models. Published.

Abualdenien, J., Schneider-Marin, P., Zahedi, A., Harter, H., Exner, H., Steiner, D., , Mahan Singh , M., Borrmann, A., Lang, W., Petzold, F., Geyer, P. . . Schnellenbach-Held, M. (2020). Consistent management and evaluation of building models in the early design stages. Journal of Information Technology in Construction, 25, 212-232. doi:10.36680 / j.itcon. 2020.013

Abualdenien, J., & Borrmann, A. (2019). A meta-model approach for formal specification and consistent management of multi-LOD building models. Advanced Engineering Informatics, 40, 135-153. doi:10.1016 / j.aei.2019.04.003

Abualdenien, J., & Borrmann, A. (2020). Formal analysis and validation of Levels of Geometry (LOG) in building information models. Proc. of the 27th International Workshop on Intelligent Computing in Engineering.

Stabinger, S., Peer, D. and Sanchez, A., 2021. Arguments for the unsuitability of convolutional neural networks for non-local tasks.

Jia, S., Jiang, S., Lin, Z., Li, N., Xu, M. and Yu, S., 2021. Neurocomputing. A survey: Deep learning for hyperspectral image classification with few labeled samples,

Sze, V., 2017. Efficient Processing of Deep Neural Networks: A Tutorial and Survey,

Brodzicki, A., Korjakowaska, J., Kleczek, P., Garland, M., & Bogyo, M. (2020). Pre-Trained Deep Convolutional Neural Network For Clostridioides Difficile Bacteria Cytotoxicity Classification Based On Fluorescence Images.

Setiawan, W., & Damayanti, F. (2019). Layers Modification Of Convolutional Neural Network For Pneumonia Detection.

Rosebrock, A. (2017). Imagenet: Vggnet, Resnet, Inception, And Xception With Keras.

Chein, J. (2020). Neural Network. What Is A Neural Network. Retrieved from https://www.investopedia.com/terms/n/neuralnetwork.asp

Pai, A. (2020). CNN Vs. RNN Vs. ANN – Analyzing 3 Types Of Neural Networks In Deep Learning.

Zaremba, W. (2015). RECURRENT NEURAL NETWORK REGULARIZATION.

Goodfellow, I., Bengio, Y., & Courville, A. (2017). Deep Learning " Efficient processing of Deep Neural Networks " (pp. 41-64).

Bargstadt, H. (2015). Challenges Of BIM For Construction Site Operations.

Manrique, J., Hussein, M., Bouferguene, A., & Nasseri, R. (2015). Automation in Construction. Automated Generation Of Shop Drawings In Residential Construction, 55.

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine Learning And Deep Learning. doi: https://doi.org/10.1007/s12525-021-00475-2

Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., & Zhu, H. et al. (2018). Machine Learning And Deep Learning Methods For Cybersecurity.

Huang, R., Pedoeem, J., & Chen, C. (2018). YOLO-LITE: A Real-Time Object Detection Algorithm Optimized For Non-GPU Computers.

Zeng, Z., Li, X., Yu, Y., & Fu, C. (2019). Deep Floor Plan Recognition Using A Multi-Task Network With Room-Boundary-Guided Attention.

Ahmed, S., Liwicki, M., Weber, M., & Dengel, A. (2011). Improved Automatic Analysis Of Architectural Floor Plans.

Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks For Large-Scale Image Recognition.

Or, S., Wong, K., Yu, Y., & Chang, M. (2005). Highly Automatic Approach To Architectural Floorplan Image Understanding & Model Generation.

Tang, S., Shelden, D., Eastman, C., Bozorgi, P., & Gao, X. (2019). Automation in Construction. A Review Of Building Information Modeling (BIM) And The Internet Of Things (Iot) Devices Integration: Present Status And Future Trends, 101, 127-139.

Sacks, R., Girolami, M., & Birlakis, L. (2020). Developments in the Built Environment. Building Information Modelling, Artificial Intelligence And Construction Tech, 4.

StandfordUniversity. (2016). ARTIFICIAL INTELLIGENCE AND LIFE IN 2030.

Sharma, N., Sharma, R., & Jindal, N. (2021). Global Transitions Proceedings. Machine Learning And Deep Learning Applications-A Vision, 2(1), 24-28. doi: https://doi.org/10.1016/j.gltp.2021.01.004

Corominas, O., Smolinska, I., & Rana, A. (2021). Image Classification With Classic And Deep Learning Techniques.

Loukas, S. (2020). Multi-class Classification: Extracting Performance Metrics From The Confusion Matrix.

Flach, P. (2019). Performance Evaluation in Machine Learning: The Good, the Bad, the Ugly, and the Way Forward. Proceedings of the AAAI Conference on Artificial Intelligence, 33(01), 9808-9814.

Saito, T., & Rehmsmeier, M. (2015). The Precision-Recall Plot Is More Informative Than The ROC Plot When Evaluating Binary Classifiers On Imbalanced Datasets. doi: https://doi.org/10.1371/journal.pone.0118432

Guan, Q., Wang, Y., Ping, B., Li, D., Du, J., & Qin, Y. et al. (2019). Deep Convolutional Neural Network VGG-16 Model For Differential Diagnosing Of Papillary Thyroid Carcinomas In Cytological Images: A Pilot Study.

Deng, M., Goy, A., Li, S., Arthur, K., & Barbastathis, G. (2020). Probing Shallower: Perceptual Loss Trained Phase Extraction Neural Network (PLT-Phenn) For Artifact-Free Reconstruction At Low Photon Budget, 28(2), 2511-2535. doi: https://doi.org/10.1364/OE.381301

Nash, W., Drummond, T., & Birbilis, N. (2018). A Review of Deep Learning in the Study of Materials Degradation.

Koylo, C., Zhao, C., & Shao, W. (2019). Deep Neural Networks and Kernel Density Estimation for Detecting Human Activity Patterns from Geo-Tagged Images: A Case Study of Birdwatching on Flickr. doi:10.3390/ijgi8010045

Shinde, S., Kothari, A., & Gupta, V. (2018). Procedia Computer Science. YOLO Based Human Action Recognition and Localization, 133, 831-838. doi:https://doi.org/10.1016/j.procs.2018.07.112

Luke, J., & Joseph, R. (2019). Impact of Image Size on Accuracy and Generalization of Convolutional Neural Networks.

Brownlee, J. (2020). Better Deep Learning (Vol. 1.8).

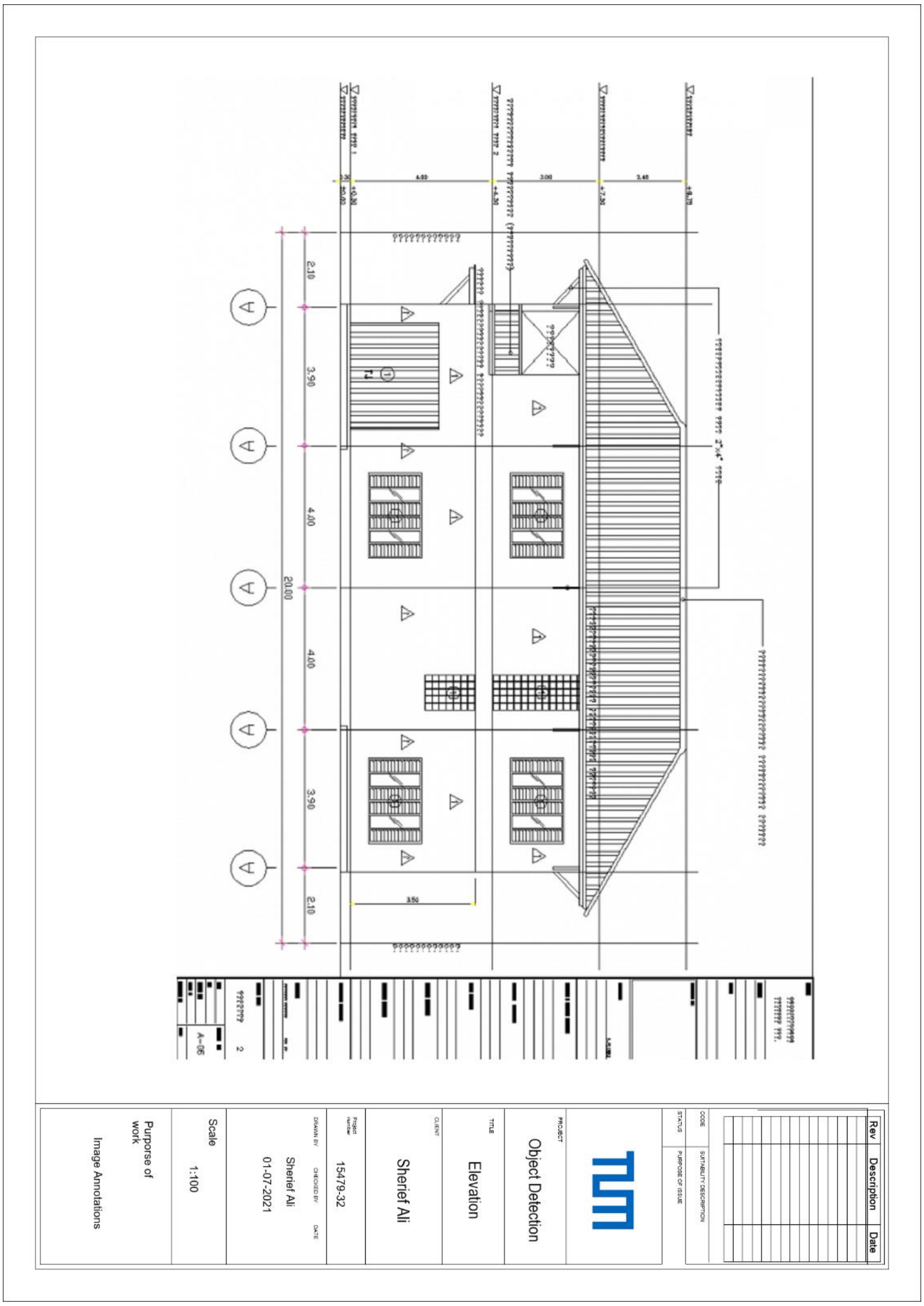
Cai, S., & Nielsen, E. (2020). Underfitting, overfitting, and the universal workflow of machine learning. In 1467014511 1061715756 S. Bileschi (Ed.), Deep Learning with JavaScript.

Rezvanifar, A., Cote, M., & Albu, A. (2019). Symbol spotting for architectural drawings: State-of-the-art and new industry-driven developments. doi:https://doi.org/10.1186/s41074-019-0055-1

Appendix

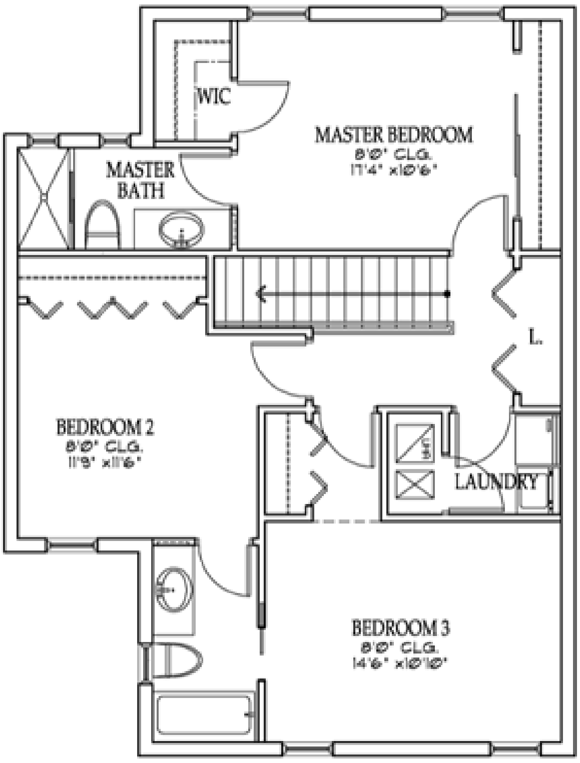
Digital Drawings

Elevation



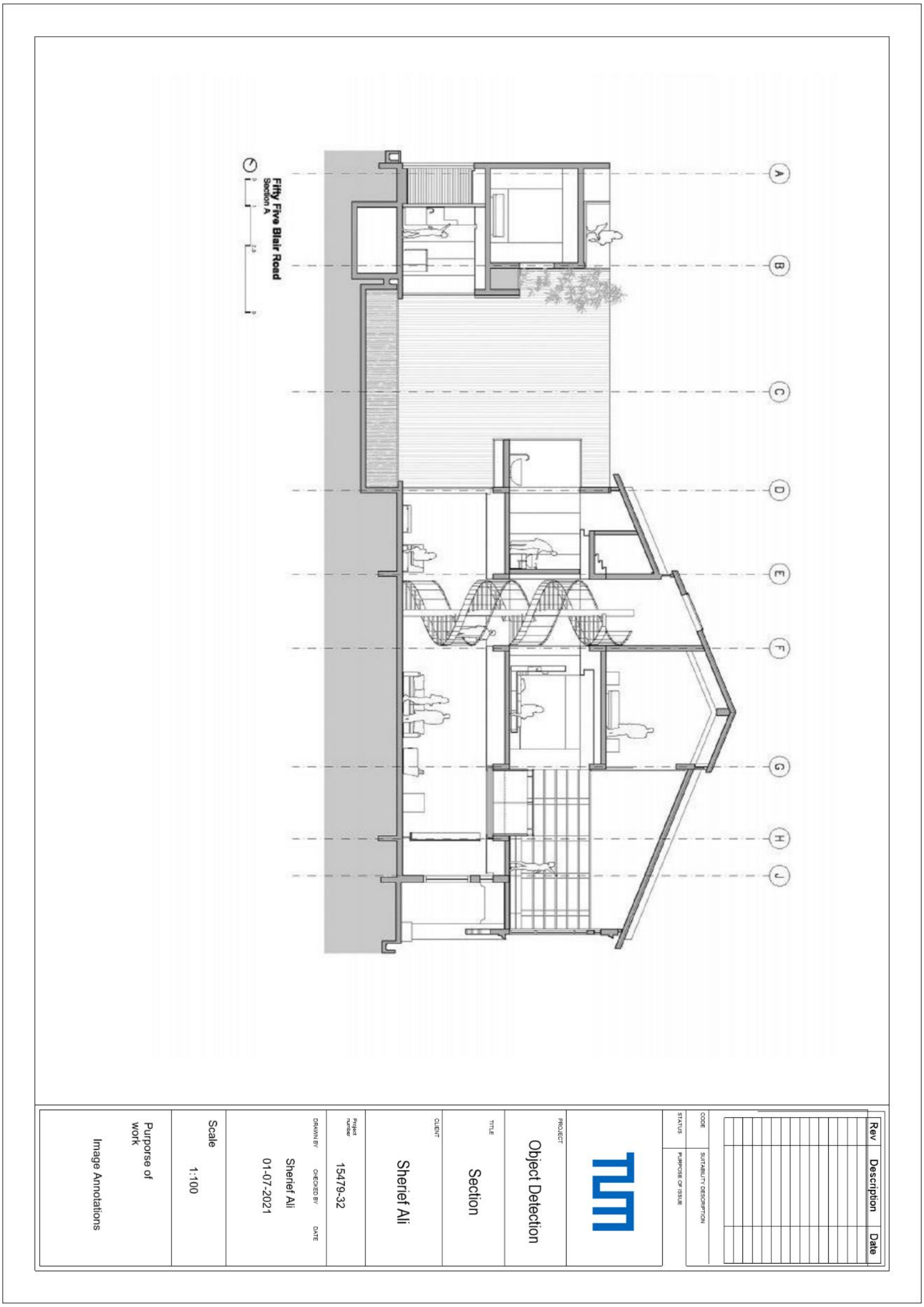
Floor plan

Projektnummer	1220-25
Projekt	Object Detection
Bauherr	Sherief Ali
Plan	Floor Plan

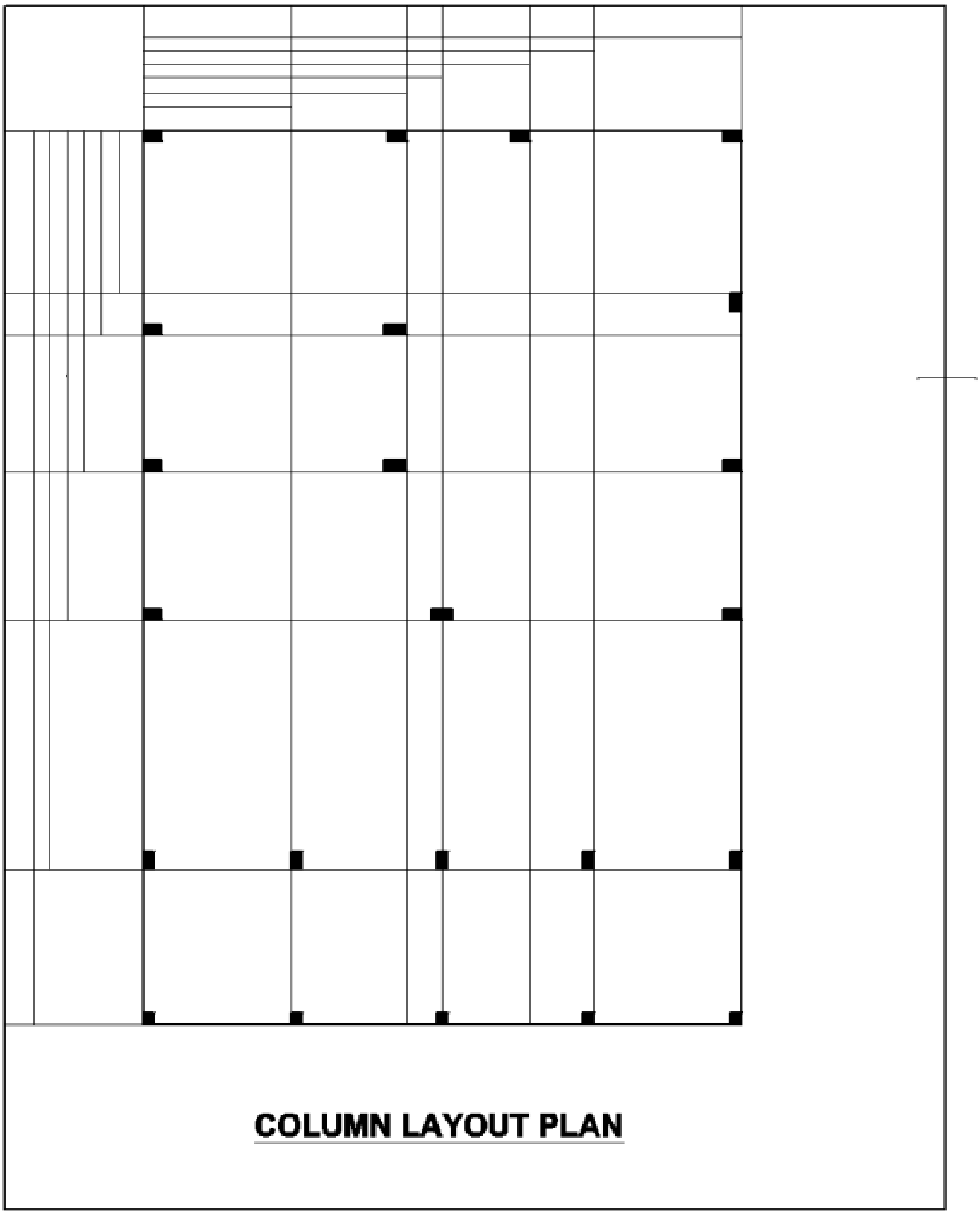


MODEL "B" SCOND FLOOR PLAN
(Unit 3,11)

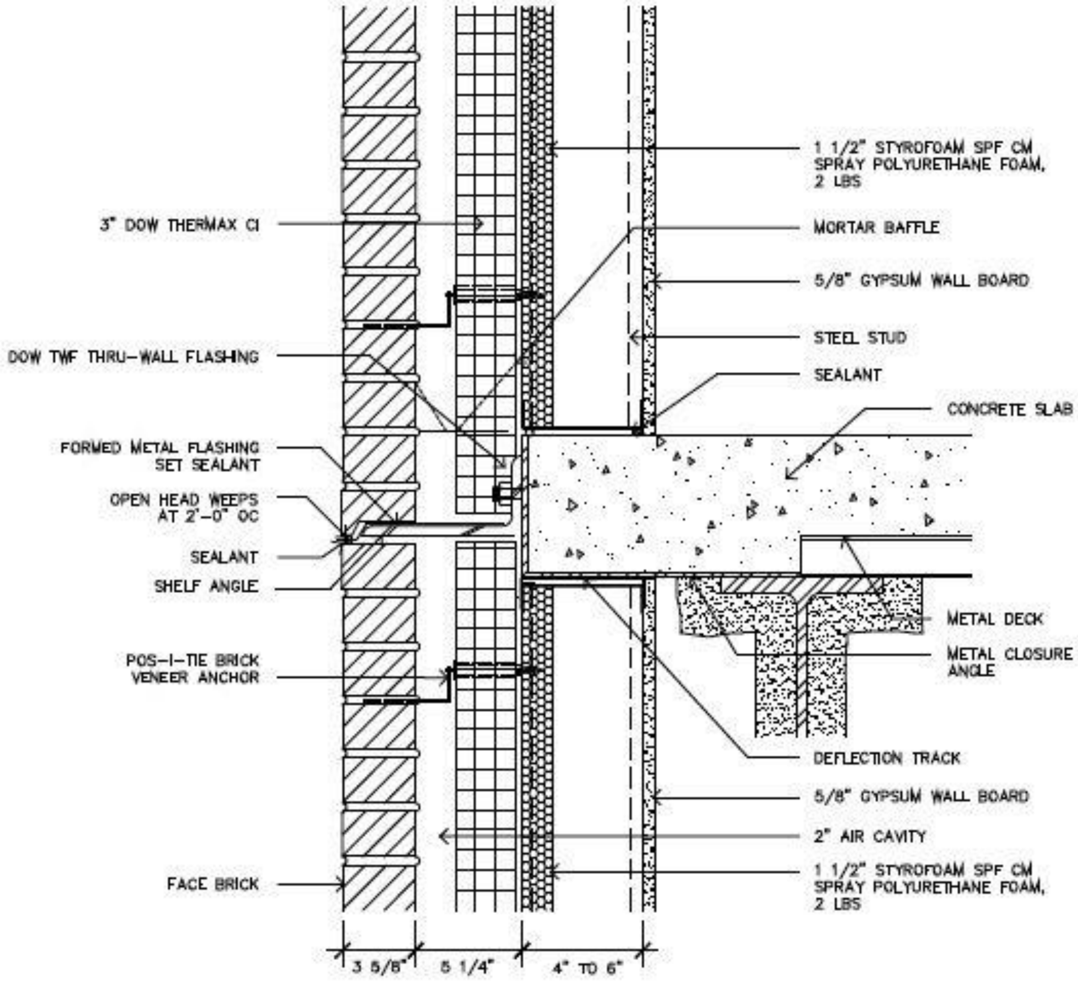
Section drawing



Column Layout



Detail Drawing

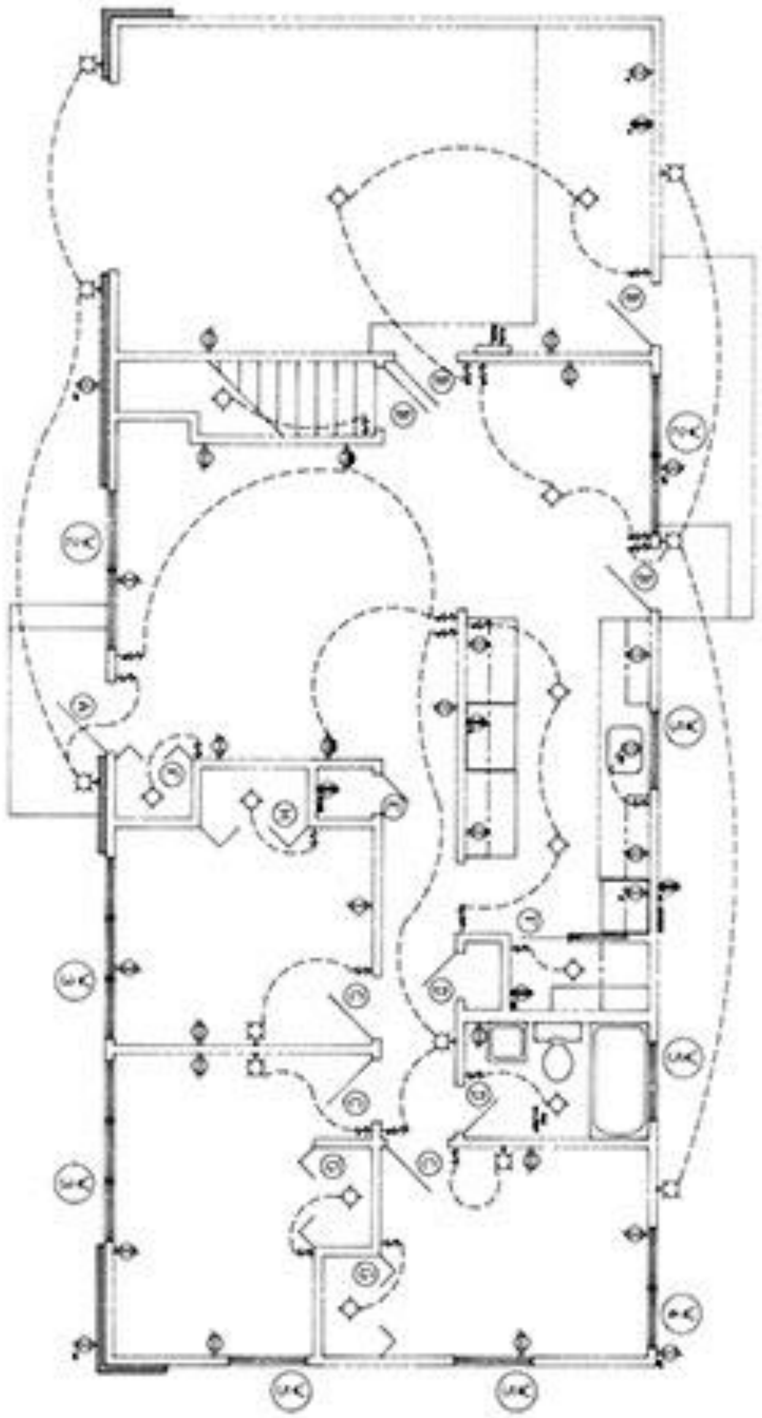


SECTION AT FLOOR SLAB

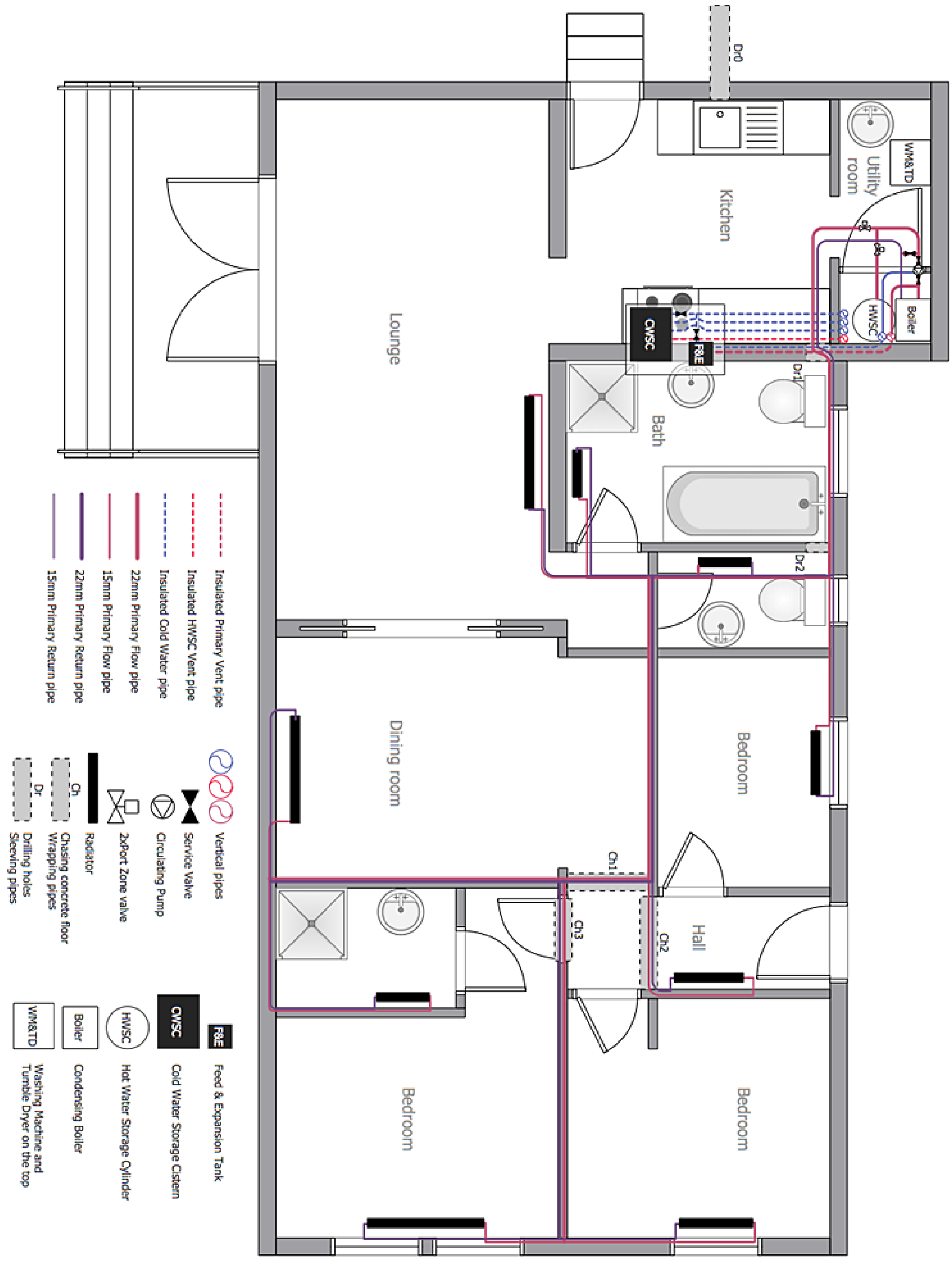
REFERENCE DETAIL: REGISTERED PROFESSIONAL TO REVIEW PRIOR TO USE

DETAIL	TITLE: BRICK STEEL STUD DESIGN D: R VALUE= 27, U VALUE=.037	SKETCH NUMBER SK-A3 4 OF 9
PACE REPRESENTATIVES INC. One Rockdale Street, Braintree, MA 02184	Date: 00/00/00 Scale: 1-1/2"=1'-0" Drawn: ---	

Electrical Plan



Plumbing Plan



Declaration

I hereby declare that I wrote the present Master's thesis on my own. Only the sources and resources expressly named in the work were used. Literally or analogously adopted ideas I have identified as such.

I also assure that the present work has not yet been used as the basis for another examination procedure.

München, 15. September 2021 Sherief Ali

Vorname Nachname

Sherief Ali

Sherief.ali@tum.de