DETECTING UNSTRUCTURED TEXT IN STRUCTURAL DRAWINGS USING MACHINE VISION

by

Jean Herfina Kwannandar

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science in Civil Engineering

Lyles School of Civil Engineering

West Lafayette, Indiana

August 2022

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Shirley J. Dyke, Chair
Lyles School of Civil Engineering

Dr. Julio A. Ramirez
Lyles School of Civil Engineering

Dr. Mohammad Reza Jahanshahi
Lyles School of Civil Engineering

Approved by:
Dr. Dulcy M. Abraham

Dedicated to Jesus Christ, my beloved parents, husband, and son.

Thank you for all the support and love.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor Dr. Dyke for the continuous support of my master study, for her patience, encouragement, and immense knowledge. Her valuable guidance supported me in my research and made this an unforgettable journey in my study.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Ramirez and Dr. Jahanshahi for their insightful comments and encouragement. Also, I would like to say thank the CV team and IISL, which helped their best to review and discuss my master thesis.

Next, I want to thank Ministry of Finance for the Indonesia Endowment Fund for Education scholarships program which supports and enables me to pursue my study in Purdue University.

Furthermore, I would like to acknowledge Naver Corp. and Mayank Kumar Singh, as fine-tuning training programming code of the CRAFT text detection implementation, which is applicable for the structural drawings, based on their works.

In addition, I would like to acknowledge Ankush Gupta et al. since their work on SynthText dataset is highly important for the predictive model training.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This focus of this thesis is the application of text detection, which is a field within computer vision, in structural drawings. To understand a structural system and conduct a rapid assessment of an existing structure would benefit from the ability to read the information contained within the drawing or related engineering documents. Extracting engineering data manually from the structural drawings is incredibly time-consuming and expensive. In addition, the variation in human engineers' experience makes the output prone to errors and false evaluations. In this study, the latest development in computer vision, especially for text detection, using large volumes of words in some structural drawings, is explored and evaluated. The goal is to read text in structural drawings, which usually has some feature noises due to the high complexity of the structural annotations and lines. The dataset consists of computer-generated structural drawings which have different word shapes and types of fonts with various text orientations. The utilized structural drawings are floor plans, and thus contain structural details which are filled with various structural element labels and dimensions. Fine tuning of the pre-trained model yields significant performance in unstructured text detection, especially in the model's recall. The results demonstrate that the developed predictive modeling workflow and its computational requirements are sufficient for the unstructured text detection in structural drawings.

1. INTRODUCTION

1.1 Motivation

The inspection of existing structures is one of the critical tasks that engineers do. This is important because our infrastructure is rapidly growing and clearly needs to be monitored to ensure structural safety, especially when a building is reaching its design lifespan. In addition, there is a possibility that future demand of a given structure will increase based on an updated design code or new designated occupancy, which makes previous design assumptions irrelevant. Furthermore, the asset management systems for the infrastructures in every country often have structural drawings in their archived document databases. If the structure was built before the implementation of digital asset management systems, the conversion of the printed structural drawings would be necessary to extract these details in an automated way. The digitalization of these structural drawings would be especially helpful when structural safety needs to be quickly assessed by an engineer in a post-hazard situation.

One of the key factors to understand the structural system is the ability to read and understand the information contained within the drawing or related engineering documents. For instance, understanding a bridge structure from the written document requires the engineers to locate the geographic location of the bridge, determine the span of the bridge, and identify the section of the girders, the dimension of the piers, and the reinforcement details of the concrete piers and piles. Also, engineers may need to further assess the bridge by modelling the geometry of the bridge, inputting the material properties, the loadings, and other relevant assumptions. However, these tasks face some obstacles that need to be resolved. First, many engineering documents that need to be reviewed during a specific project. Extracting engineering data manually from the structural drawings is incredibly time-consuming and would require a large number of man-hours. This is prohibitively expensive in most developed countries. Second, the experience of human engineers varies considerably from person to person, and this may add uncertainty.

To overcome these challenges, we aim to exploit computer vision for information extraction. Computer vision is one form of Artificial Intelligence which enables computers to process visual information and recognize the pattern of environments. At the present time, there are large numbers of images available which can be used to improve computer vision algorithms. Furthermore, the development of computer vision is strengthened by the new development of new GPU hardware, so that neural network training and utilization is becoming faster and cheaper. According to Lavrentyeva (2021), computer vision is used widely in retail & ecommerce, education, healthcare, fitness & sports, agriculture, manufacturing industries, and mining industries. Furthermore, the application of computer vision has been transferred to other new fields, especially to civil engineering problems, which were conducted by Yeum et al (2015), Spencer Jr. et al (2019), and others. Automation provided by modern computer vision into the repetitive process of information extraction is simply one example this. However, we have not found publications that explain the application of computer vision to the unstructured text detection in structural drawings.

The focus of this thesis is to examine the potential for applying the latest developments in text-detection algorithms for the purpose of reading and extracting the information contained within structural drawings. Specifically, the goal is to detect and read, or interpret the text related to the dimensions provided in the drawings that are associated with structural components. The method is applied to computer-generated structural drawings with various font styles and shapes. The results demonstrate that a predictive model can detect the unstructured text with good performance.

1.2 Objectives and Contributions

In this study, the research objective is to utilize the latest development of computer vision, especially in text detection, using a ground truth training set containing large volumes of words from structural drawings. This method will enable an automated interpretation of the engineering information which is available in structural drawings or engineering documents. By utilizing the latest technology of text detection, the computer is capable of detecting text from the structural drawings. Structural drawings have high visual complexity due to the integration of graphics related to the structural elements, plus those for all of the annotations and lines. Promising applications of this method is for digitalizing the information from an image, for instance, structural modelling for assessment, bill of materials calculation, and generating a summary of the information about the drawings.

The contribution of this research is applying modern computer vision to a new purpose to transform the slow conventional structural assessment with large volumes of structural drawings and engineering documents collected from the real-world design output into faster, reliable, and accurate future structural assessment.

1.3 Scope of Work

The text detection method is applied to real-world structural drawings, and the results demonstrate the potential of this method. In Chapter 2, a literature review is provided explaining the historical development of computer vision and text detection methods. In Chapter 3, the methods adopted for this work are described and a rationale is provided for why these methods are most appropriate for the work included in the thesis. In Chapter 4, the network architecture used for detecting unstructured text in structural drawings is described. In Chapter 5, the dataset is described, and the results are provided, with lessons learned along the way regarding overcoming the challenges in text detection. In Chapter 6, the conclusions of the study are discussed and the suggestions for the future research are shared.

2. LITERATURE REVIEW

2.1 Structural Drawings

In building construction projects, there are five crucial phases, which are: initialization, planning, execution, controlling, and project closing. Each of the phases requires communication media among stakeholders (owners, professional designer, and construction contractor) for easy implementation in the field. Moreover, printed documents, such as contracts and drawings, would be required in this communication. Furthermore, the drawings are different in each phase. In this research, structural drawings are utilized for the case study.

Structural drawings must be made developed before a building is constructed, based on the architectural drawings, and are required to satisfy local design code and laws. These drawings explain the general notes, set of plans, building elevations, building sections, and details of the building structure. In the past, these drawings were drawn manually by engineers. Nowadays, the development of computer tools enabled a revolution in the construction sector, especially in structural drawings. The number of drawings depends on the structure’s complexity. Therefore, digitalization of the complex and sets of numerous structural drawings would be important to support their rapid interpretation by the engineers.

2.2 Deep Learning

The data and information in this technology era is significantly increasing over time. However, humans are limited in the speed and consistency with which they can process these large volumes of data. Therefore, enabling computers to assist humans in interpreting and making decisions based on such data could be beneficial. Realizing the high potential of Artificial Intelligence (AI) in the future, many researchers consider how to find and use the best algorithms to revolutionize their field. This approach requires the computer to perform a type of regression on a given pattern and train a model such that it can provide the correct prediction for new data inputs. This topic is called machine learning. Moreover, this field is supported by the exponentially increasing computational power of the computer, which is why machine learning is growing rapidly and gaining traction in various fields.

Currently, machine learning has evolved to be able to interpret the complex and high-feature patterns, giving the appearance that it can understand images and data. This class of machine learning requires multiple layers of artificial neural networks, which are also complex and have high nonlinearity. When there is a really large number of parameters involved in the calculation, it is considered to be deep learning. Furthermore, the complex tasks, such as computer vision, natural language processing, language translation, can be solved by the deep learning architecture and the result of the prediction is comparable, even exceeding, the human performance doing similar tasks.

The methods for deep learning training of the typical network architecture are:

- Determining the proper deep learning architecture for the case study.
- Prepare the datasets for the model training and testing purposes.
- The datasets are inputted to the model. Then, they are processed with the multi-layer calculation to produce the prediction or output.
- In model training, the model compares the calculated output to the ground-truth output, which is based on the labelled data. This comparison is quantized as loss function. In order to achieve the best prediction, the model has to minimize the loss function. The error of the output is converted to the gradient which is backpropagated inside the model to generate new weight to the hidden layer.
- In model testing, the model calculates the output with different datasets and checks the performance of the model. This is required because the model needs to be ensured that the overfitting does not happen during the model training.

There are several types of the deep learning architectures, which are explained below.

2.2.1 Feedforward Neural Network (FNN)

Sandberg et al (2001) explained that feedforward neural network consists of multiple hidden layers connected in one direction. This artificial neural network does not form any loops between the layers. The assumption of the input is that all raw information is considered important and needs to be connected to the hidden layer.

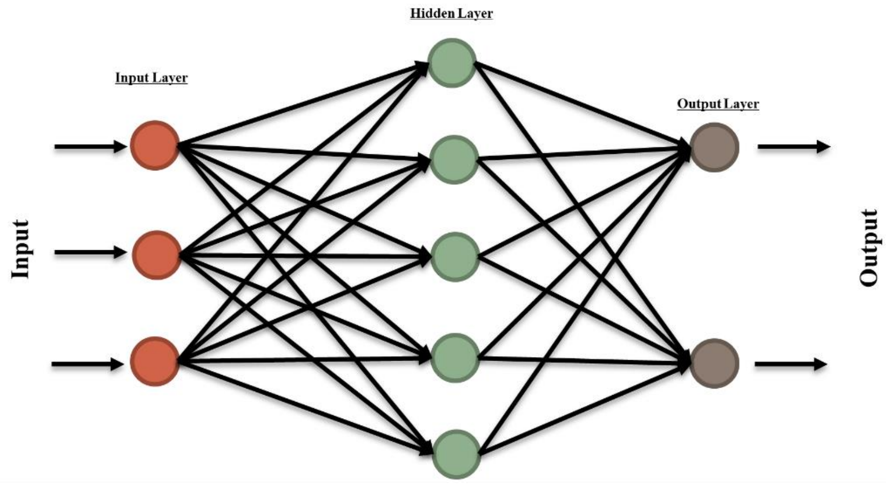


Figure 2.1. Feedforward Neural Network Typical Architecture

2.2.2 Convolutional Neural Network (CNN)

According to Saha (2018), a convolutional neural network is formed by the set of convolution kernels. These kernels are then applied to the certain raw information or processed input data and transform the previous signal into a new calculated region value. This maximum or minimum value of the region can represent the new output signal, which is processed further. Usually, the convolutional kernel outputs are then connected to the fully connected layers, which is similar with the FNN architecture, to classify the output prediction

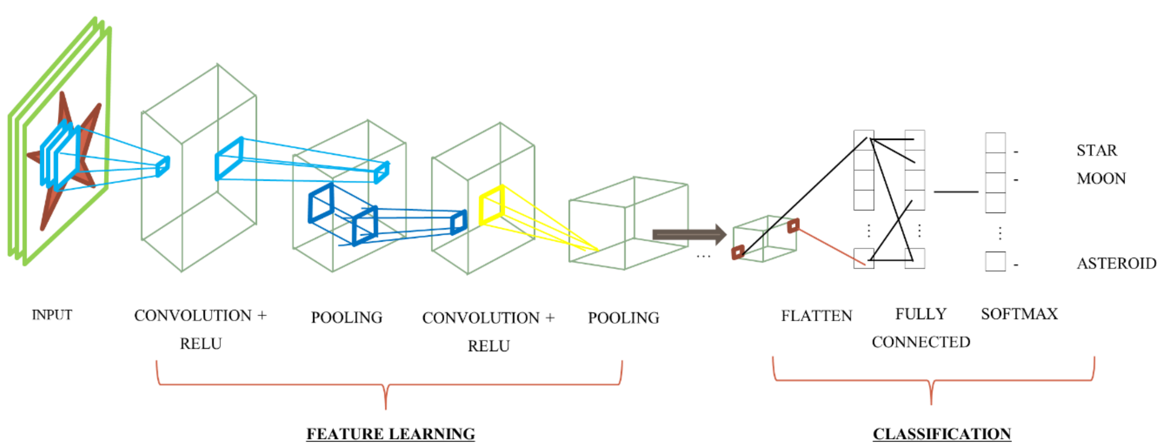


Figure 2.2. Convolutional Neural Network Typical Architecture

2.2.3 Recurrent Neural Network (RNN)

According to Biswal (2022), recurrent neural networks consist of multiple hidden layers and form some loops. It means that the output of a hidden layer is being considered as the input of the previous hidden layer. Therefore, it reflects the two-directional signal in this network. RNN is important for the sequencing tasks and is usually considered as the memory of the network.

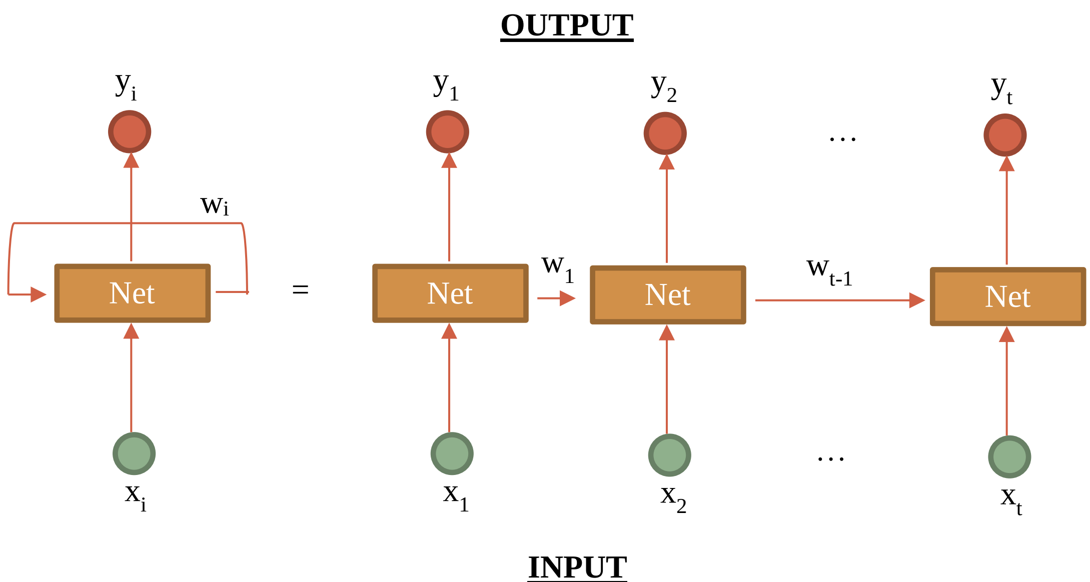


Figure 2.3. Recurrent Neural Network Typical Architecture

2.2.4 VGG16

Very Deep Convolutional Network for Large-Scale Image Recognition, which is popularly known as VGG, is designed to improve a number of parameters of the architecture and increase the depth of network by adding some convolutional layers capable of utilizing extremely small (3x3) convolutional filters in all layers. The network architecture was proposed by Simonyan et al (2015)

Furthermore, the architecture configuration consists of six types depend on the weight layes, such as: type A (11 weight layers), type A-LRN (11 weight layers), type B (13 weight layers), type C (16 weight layes), type D (16 weight layers), type E (19 weight layes). Each type of ConvNet Configuration is composed of a convolutional layer 3x3 (except for type C that has convolutional layer of 1x1), max-pooling, and three fully connected layers.

Focusing on Type D which has 16 weight layers, consisting of 5 parts, i.e. the architecture in layer 1 and 2 have a 64 channel 3x3 kernel. Next, layer 3 and 4 have a 128 channel 3x3 kernel. After that, layers 5, 6, and 7 depend on the convolutional layer of the 256 channel 3x3 kernel. Layers 8, 9, and 10 composed by convolutional layer 512 channel of 3x3 kernel. Then, layers 11, 12, and 13 have a convolutional layer of 512 channels of 3x3 kernel. Every part of the convolutional layer is followed by one max-pooling that is performed over a 2x2 pixel window, with stride 2. Afterwards, there are 3 fully connected (FC) layers which are similar with all networks. Final layer of this architecture is the Soft-max layer. The VGG16 layer with input image dimension of 224 x 224 x 3 is explained in Table 2.1. VGG16 Layer Explanation

Table 2.1. VGG16 Layer Explanation

No.	Convolution	Output Dimension after Convolution + ReLU	Pooling	Output Dimension
1.	Layer 1 and 2	224 x 224 x 64	max pool stride = 2, size 2 x 2	112 x 112 x 64
2.	Layer 3 and 4	112 x 112 x 128	max pool stride = 2, size 2 x 2	56 x 56 x 128
3.	Layer 5, 6, and 7	56 x 56 x 256	max pool stride = 2, size 2 x 2	28 x 28 x256
4.	Layer 8, 9, and 10	28 x 28 x 512	max pool stride = 2, size 2 x 2	14 x 14 x 512
5.	Layer 11, 12 and 13	14 x 14 x 512	max pool stride = 2, size 2 x 2	7x7x512

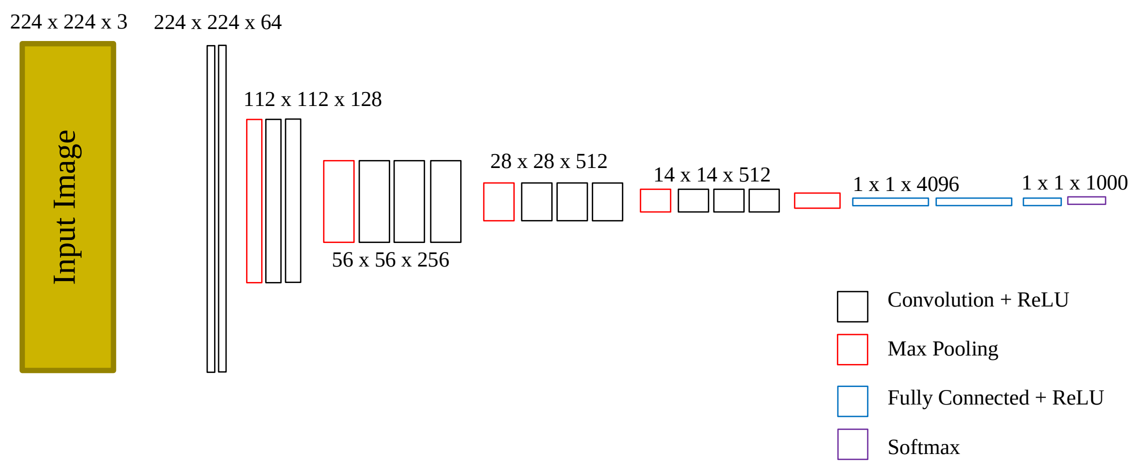


Figure 2.4. VGG16 Network Architecture with 224 x 224 x 3 Image Dimension

2.2.5 U-Net

Ronneberger et al (2021) introduced U-Net, which is a convolutional network that is based on encoder and decoder networks. This network can be trained end-to-end from very few images and outperforms the best method on the ISBI challenge for segmentation of neuronal structure in electron microscopic stacks.

In addition, U-Net is inspired by the fully convolutional network which has been modified by adding successive layers. The function of this layer is to replace the pooling operator with the up sampling operator, which has a large number of feature channels that can result in a higher resolution output.

The network architecture of U-net is divided by the contracting path and expansive path. The function of the contacting path is to process the input with 3x3 convolutions, rectified linear unit (ReLU), 2 x 2 max pooling with stride 2 for down sampling. Besides, the expansive path is using the 2 x 2 convolution for up-sampling the future map, 3 x 3 convolutions and ReLu. The total network has 23 convolutional layers.

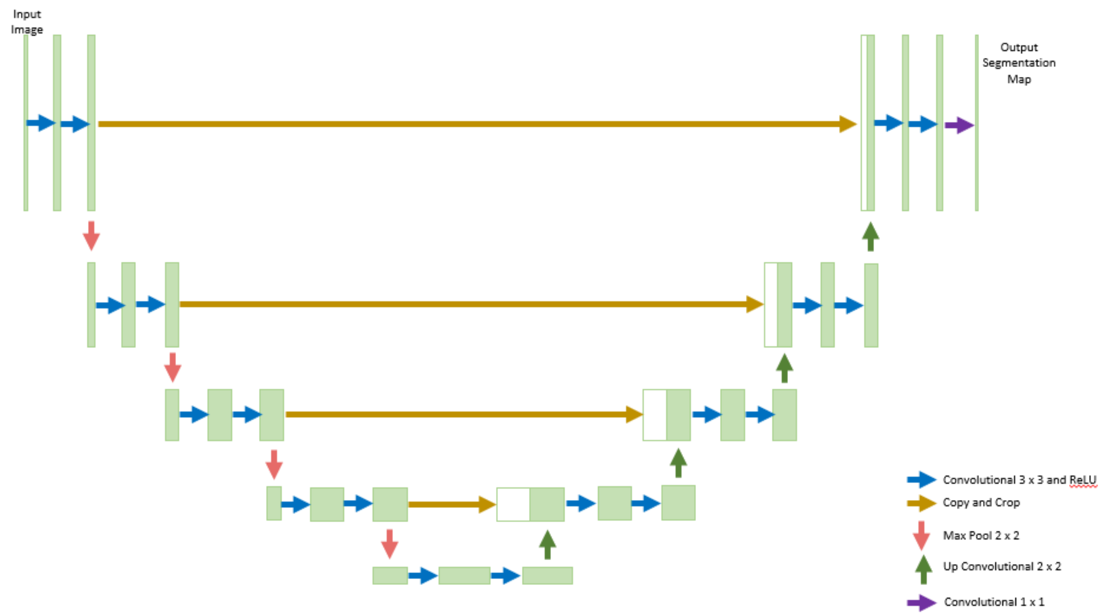


Figure 2.5. U-Net Network Illustration

2.3 Computer Vision

Computer vision is a form of artificial intelligence and is used to enable the computer to gain a high-level of information from digital images or videos. This includes information about the content of scenes, or the location of objects in an image, for instance. With this information extracted, the computer can give the appearance that understands the image, and that understanding helps it to make relevant decisions. According to Ferreira et al (2021), this technology has strong potential to assist humans in decision making based on computer visual perception. There are several tasks that can be automated by the computer using the computer vision method, such as object detection and recognition, text detection and recognition, content-based image search, optical character recognition, facial recognition, and motion analysis, scene reconstruction, image restoration, and so on. According to Lavrentyeva (2021), computer vision is widely used in a growing set of industries, ranging from manufacturing to medical diagnostics.

According to Huang (2009), research was conducted by MIT which pioneered computer vision. It has been about extracting 3D geometry from a structure. This research was the foundation of computer vision algorithms nowadays, including the extraction of the edges from images, determination of the lines, and object segmentation. The performance is increasing because the performance of deep learning networks is evolving and surpassing the prior methods. In addition, increasing numbers of labelled databases such as ImageNet, which was introduced by Deng et al (2009), and the development of the necessary technologies will also accelerate computer vision research.

Fei-Fei Li et al (2017) categorized computer vision basic tasks based on their function:

1. Image Classification

   This task is enabling the predictive model to classify an input image based on the type of predefined categories. It usually has a single category for the output. For instance, the model can differentiate the hand-written words and the computer-printed words based on the features and shapes of the input words.

2. Localization

   This task is enabling the predictive model to determine the location of the classified single object by drawing a bounding box around the object. For instance, if the hand-written word is detected in an input image, the bounding box is drawn around the highlighted words.

3. Object Detection

   This task is enabling the predictive model to determine multiple classified objects and their location inside an input image by drawing the bounding boxes. It is similar with image classification + localization, but it is applied for more than one object inside an input image. For instance, if there are hand-written word and computer-printed worda exist in an input image, the model can detect both, classify them as different categories, and highlight them with bounding boxes.

   This task is performed by suggesting the proposed region of interest for each object, which requires a more complex model compared to the image classification + localization predictive model.

4. Semantic Segmentation

   This task is enabling the predictive model to assign each pixel of the input image by a pre-determined categories label. This task is similar with object detection task, but it is highlighted as being more detailed by annotating the pixel of interest. For instance, if a hand-written word and computer-printed word exist simultaneously in an input image, the model can detect both, classify them as different categories, and highlight the words by the pixel location, usually with distinct colors.

5. Instance Segmentation

    This task is enabling the predictive model to assign each pixel of the input image by the pre-determined categories label and differentiate between objects. It means that although there are multiple objects that have same categories label, the predictive model will give some different index number to those objects. For instance, if there are two hand-written words in an input image, the model can detect both, classify them as different categories, and highlight the words by the pixel location, and label the words with different label, usually with distinct colors.

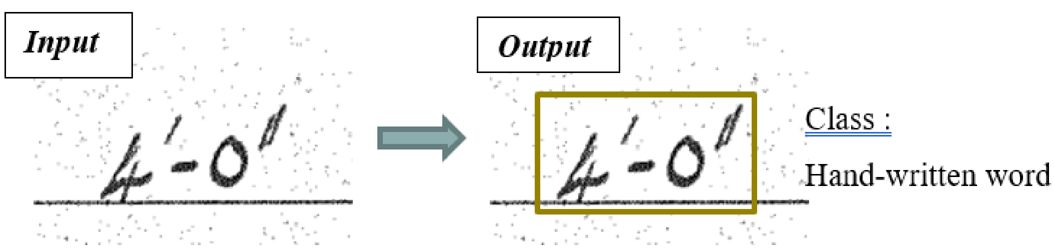


Figure 2.6. Image Classification + Localization

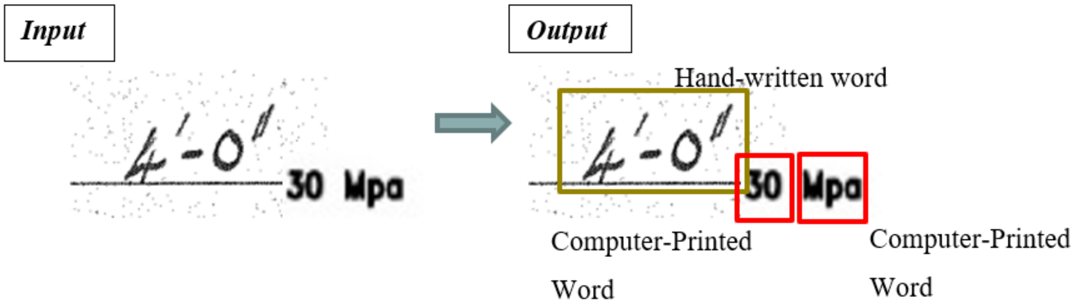


Figure 2.7. Object Detection

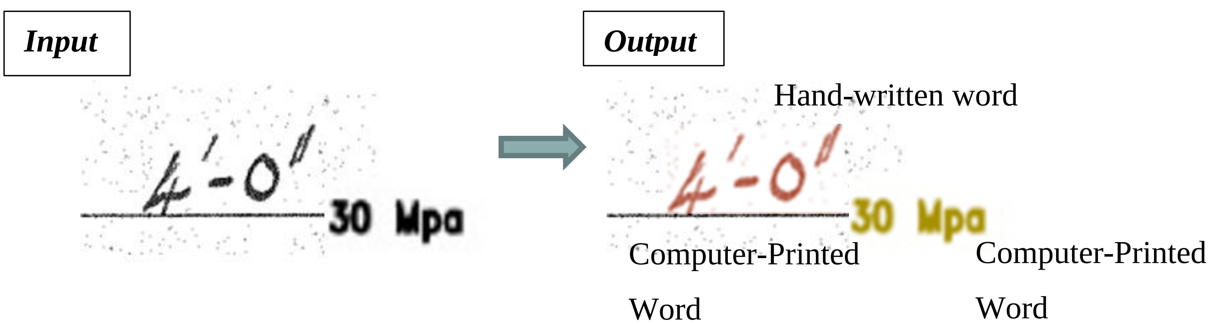


Figure 2.8. Image Segmentation

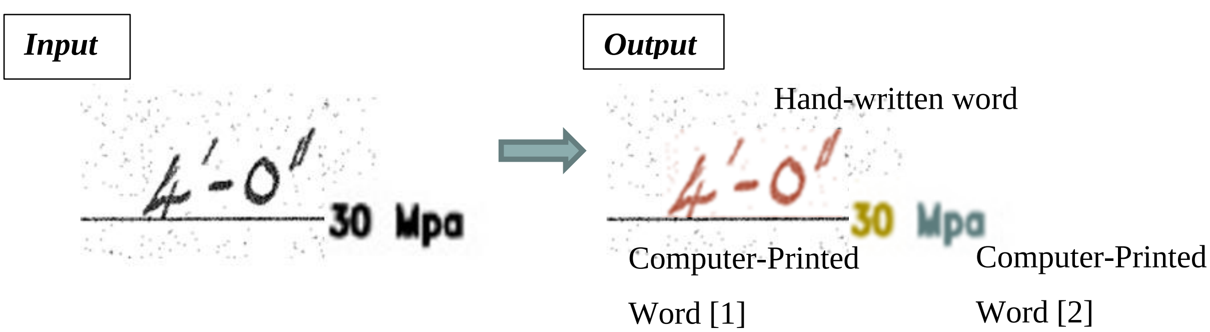


Figure 2.9. Instance Segmentation

Furthermore, this growing field of deep learning algorithms requires large image datasets to study and learn the text pattern, and then test and validate them until the performance of the predictive modelling is sufficient. Convolutional neural networks (CNN) are heavily utilized to transform the image raw data into meaningful value with less hyperparameters as compared to other deep learning networks.

2.4 Scene Text Detection

In human history, text plays an important role in written communication. Therefore, automation of text detection would enable quite a few methods for assisting humans. Many aspects can potentially assist human activities, such as image search, language translation, navigation of robots, and automation of manufactures.

According to Yao et al (2016), there are several challenges that needs to be solved, which are:

- Text Diversity and Variability in Natural Scenes
- Complex Background Noise or Inference
- Poor Quality of Image

In order to tackle these problems, synthetic word databases were generated by Gupta et al (2016) in front of various natural scene images with different orientation and distortion. Based on this huge database, the development of new algorithms for text detection is evolving.

In the past before deep learning was highly utilized, the algorithm trend of scene text detection was bottom up, where the manual-crafted features were significantly utilized, for instance, MSER or SWT, as a primary building component. On the other hand, the present deep learning utilization at text detection is popular by using object detection / segmentation algorithms, like Faster R-CNN, FCN, or SSD.

There are various types of text detector algorithms which are explained by Baek et al (2019). These include:

- Regression-based text detectors

Utilizing box regression from object detectors in text detection has been implemented. Different from the regular object in general, the texts are usually illustrated in irregular shapes with various dimensions. TextBoxes, DMPNet, and Rotation-Sensitive Regression Detector (RSDD) has been implemented to detect various text shapes, incorporating quadrilateral sliding kernels, and enable the rotated texts by actively rotating the kernels. However, there are major limitations to capture all possible text shapes when implementing this method.

- Segmentation-based text detectors

  Detecting texts at a pixel level by using segmentation has been proposed. The proposal starts from an area bounding approximation of the texts, which can be found in Multi-Scale FCN, Holistic-Prediction, and PixelLink algorithms. In addition, utilizing the attention module to increase text-related area by reducing the background interference has been implemented using the SSTD algorithm. Nowadays, the text can be detected by text region prediction and its centerline together with the geometrical properties by implementing the TextSnake algorithm.

- End-to-end text detectors

  End-to-end text detectors algorithms have been proposed to simulate the text detection and text recognition simultaneously. This problem is treated as the semantic segmentation problem and trained on the neural network in an end-to-end basis. FOTS, EAA, and Mask Text Spotters are some of the algorithms which implemented this method. However, the words can be differentiated with spaces, meaning, or color and word segmentation cannot be strictly defined. So, word annotation dilutes the ground-truth meaning for regression and segmentation problems.

- Character-level text detectors

  MSER has been proposed to predict text block candidates and detect individual characters. However, it has a drawback if the images have low contrast, are highly distorted, or exhibit light reflection. In addition, generating prediction maps of the characters requires character level of annotation. In order to tackle this problem, WordSup has proposed a weakly supervised algorithm to train the neural network. However, due to the camera point of view, text detection is vulnerable to perspective shapes. Also, the number of anchor boxes and the sizes are limited.

One of the crucial performance metrics of predictive modelling in scene text detection is *precision*. Precision is defined as the ratio between the number of true positives and the summation of the number of true positives and false positives. This metric reflects the number of correct positive predictions made. If the precision is low, it means that many text detection predictions are not at the ground truth labelled area.

However, the performance of predictive modelling in scene text detection not only depends on the precision or accuracy, but also the *recall*. Recall is defined as the ratio between the number of true positives and the summation of the true positives and false negatives. This metric reflects the number of the correct positive predictions compared to all positive predictions that are made. If the recall is low, much of the text goes undetected by predictive modelling.

Those metrics above need to be maximized. For representing the precision and recall as a single value, the F-measure score is introduced by van Rijsbergen (1979). The definition of F-measure is:

$$F - Measure = \frac{2 \times Precision\ \times Recall}{(Precision + Recall)}$$

This measure represents the performance of the predictive modelling in a balanced way. Low value at one of the metrics produces a low score of the F-measure.

2.5 Character Region Awareness for Text Detection

Previous work toward scene text detection development was significant and showed good performance. The method is training the model to understand the text inside the image and make predictions of the word-level bounding boxes. However, there is a difficulty to predict highly distorted words. The highly distorted words usually have varied font style or thickness, and arbitrary shapes. The number of the datasets with high-level annotation at the character level is very rare. In addition, doing a high-level annotation at the datasets is time consuming and requires lot of effort.

Baek et al (2019) introduced Character Region Awareness for Text detection (CRAFT) which can calculate the character region score and affinity region score. These scores are mapped into a contour map and then used to make the prediction of the bounding boxes. The character regions score represents the concentration value of the characters inside the texts and the affinity score is utilized to group the characters into words. This text detection method is popular and one of the best current algorithm based on benchmark datasets, such as ICDAR and MRSA. Due to the limited character-level annotation dataset, weakly supervised learning can be conducted by generating pseudo ground truth at word-level annotation datasets using cropped words to estimate the characters position and their affinity. It requires two types of data for the weakly supervised training; quadrilateral annotations for cropping word images and the number of the characters inside the words. It is important to input the quadrilateral coordinates and the transcribed words inside the third-app application for image labelling.

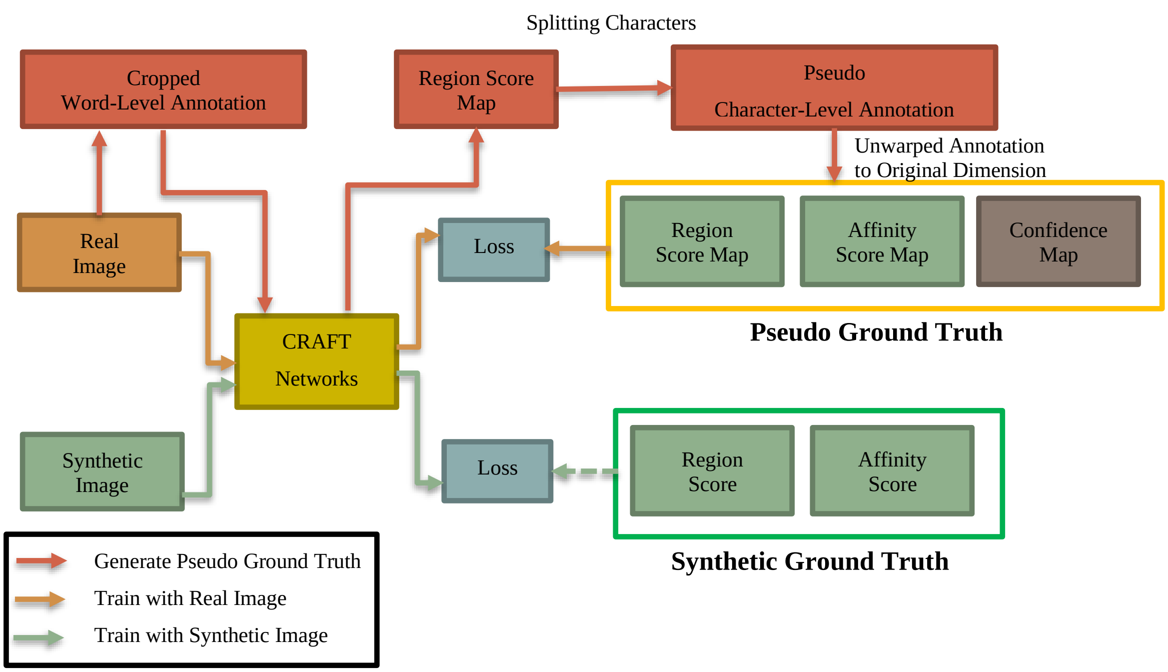


Figure 2.10. Overall Training Workflow in CRAFT Algorithm

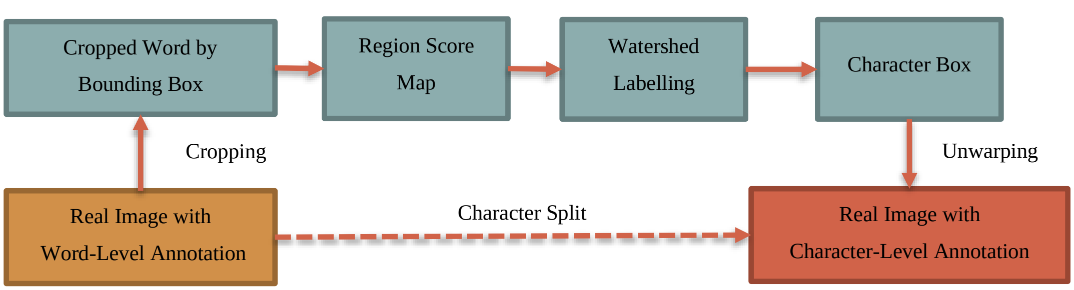


Figure 2.11. Conversion of Word-Level Annotation to Character-Level Annotation

Once word-level annotation existed, Baek et al (2019) proposed the implementation of character split algorithm for generating character-level annotation. The image is cropped using word boxes that are labelled in advance, and a region score is generated using the existing predictive model. Then, the region score heatmap is processed by watershed algorithm in order to determine the boundary of possible adjacent characters. The watershed algorithm, which was introduced by Beucher (1994), is an algorithm in image segmentation to determine the boundary of the object. The methods consist of these steps:

- Approximate the object using the color conversion and threshold algorithm
- Find the sure background area by using dilation algorithm and foreground area by using distance transform and threshold algorithm
- Determine the unknown region-based subtraction of the background area and foreground area
- Determine the marker of the sure object
- Expand the marker region by using watershed code to estimate the boundary of the sure objects.

Based on the largest dimension of each sure character position, the character boxes are generated by using the object boundaries produced by the watershed algorithm. Then, the character boxes are unwarped or transformed to the original image and the character-level annotations are generated.

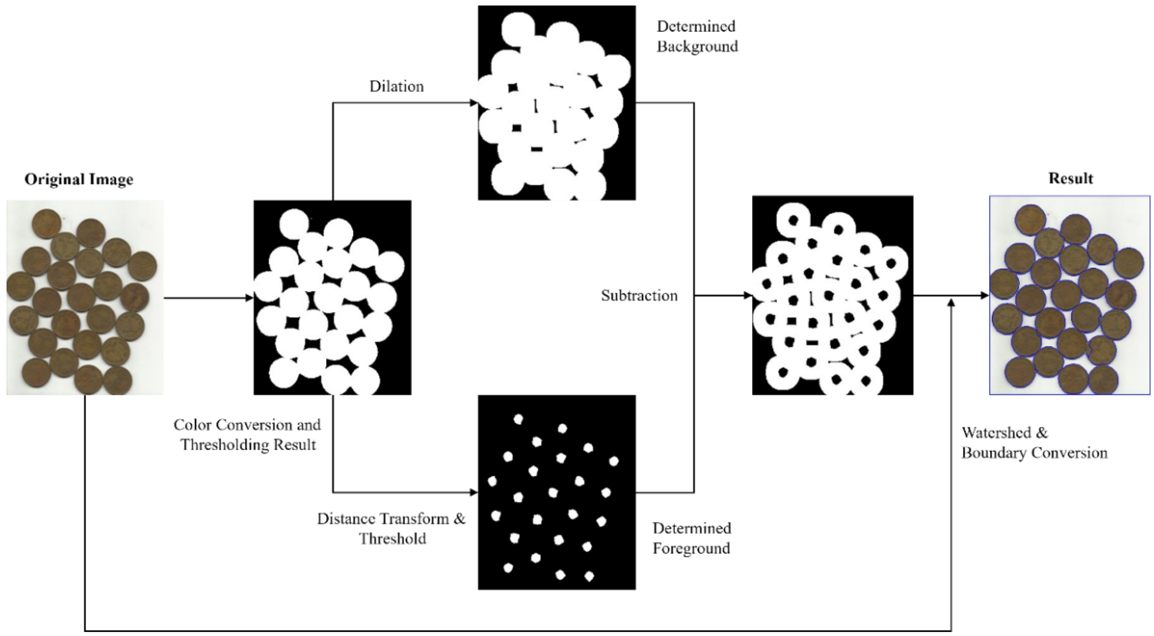


Figure 2.12. Watershed Algorithm

The real or pseudo ground truth heatmap score generation is based on the character box dimension, which is illustrated in Figure 2.13. The character boxes are explicitly obtained by strong supervision labelling or watershed algorithm and the affinity boxes are generated based on the center of the top and bottom triangles made by diagonal intersections of the character boxes. Then, the rectangle 2D Gaussian contour is transformed to each bounding boxes inside the image. So, region score map and affinity score map of each input image can be obtained.

Baek et al (2019) proposed the architecture of the CRAFT algorithm based on a fully convolutional network based on VGG-16 with batch normalization. The skip connections are modelled at the decoding part, which is similar to U-Net. The outputs of the model are the region score and the affinity score. The network architecture is displayed on Figure 2.14. CRAFT Network ArchitectureFigure 2.14.

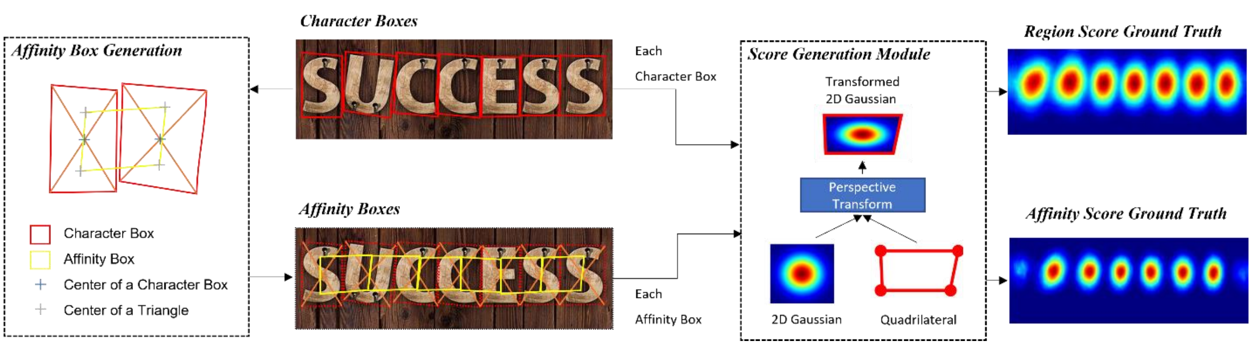


Figure 2.13. Ground Truth Generation Procedure

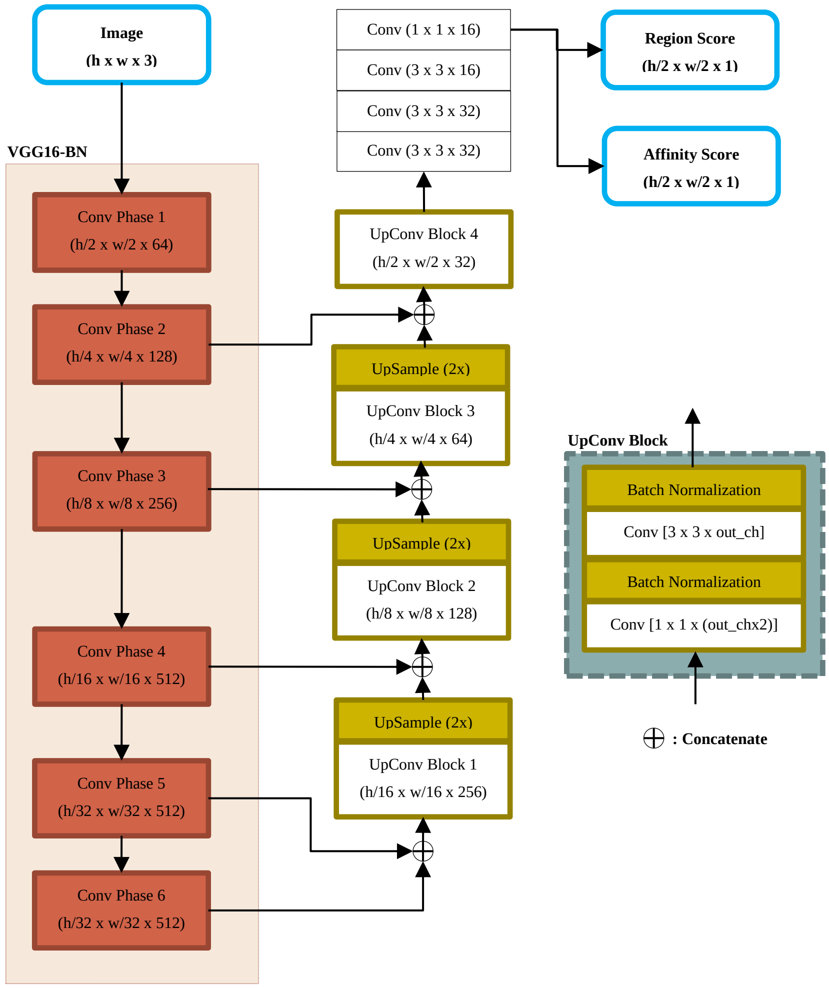


Figure 2.14. CRAFT Network Architecture

Region score represents the probability that a given pixel is at the center of the character, and the affinity score represents the probability that a given pixel is at the center of the space between characters. If these scores are generated, it would be easier to determine the text and make the bounding box prediction.

Some SynthText datasets, which are utilized in strong supervision, are still required to maintain the knowledge of the good scene text detection prediction. The SynthText dataset is widely used for scene text detection. Gupta et al (2016) proposed the synthetic text dataset, which consists of 800,000 images with the 8,000,000 synthetic words, complete with the text values, word bounding boxes, and character bounding boxes. The words are also well augmented with different shapes, color, rotation, and distortion. In addition, they are also located in natural scene images, which are helpful for giving the background noise in the training and testing.

Batch normalization method, proposed by Ioffe et al (2015), is utilized in an up-convolutional block to ensure better performance in training deep neural networks. There is a phenomenon in neural network training called internal covariate shift. Internal covariate shift is defined as the distribution changes in network activations due to the network parameters change in training. This shift requires the training to utilize lower learning rate for avoiding the gradient explosion and careful parameter initialization. The method of batch normalization is normalizing the input of the mini batch by calculating the mean and variance of the mini batch, then the normalized input is scaled and shifted by the trainable parameters. The utilization of batch normalization can reduce the overfitting problem without depending too much to the dropout algorithm, increase learning rate so the training would be faster, make the stochastic training better by shuffling the training dataset more thoroughly, and reduce the distortions existed in the images.

In order to do better stochastic gradient-based optimization in neural network training, Kingma et al (2017) introduced the Adam Optimizer algorithm. The optimizer is using exponential moving average gradients and squared gradients. Then, the optimizer modifies their biases for updating the parameters. So, the updated parameters are calculated based on the determined learning rate and modified gradients. The method has some advantages, which are:

- Efficient computation
- Fewer memory requirements
- Recommended for the large number of parameters training.
- Robust for the wide range of the non-convex optimization problems

2.6 Application to Structural Drawings and Current Challenges

Based on the latest work in text detection, it should be feasible to achieve unstructured text detection in images of structural drawings. Having the ability to automate and extract the information from the structural drawings would be beneficial.

However, fine-tuning of the model will likely be required due to the presence of complex structural drawings features which have not yet been included in the training datasets. Therefore, it can possibly produce low prediction and recall value.

3. RESEARCH METHODS

The methods in this research consist of 8 main steps, which are:

1. Image Labelling
2. Image Pre-processing
3. Dataset Separation
4. Image Rotation for Data Augmentation
5. Dataset Generation
6. Model Initialization
7. Model Training
8. Model Testing

These main steps are illustrated in Figure 3.1 and explained further in the following subsections.

3.1 Image Labelling

Texts inside the structural drawings are marked inside of a third-party application for the image labelling, such as LabelBox. The position of each labelled word is exported into a .json file to be processed further.

3.2 Image Pre-processing

The structural drawings usually have a high resolution as compared to more typical images and are sensitive to dimension resizing. Therefore, image cropping with certain resolution is utilized so the images are compatible with existing networks. Overlapping the cropping position is important to make sure there are no truncated words, which would become a disturbance for model training and model testing. Furthermore, deleting cropped words when they occur inside the cropped image is important to minimize the false negative of the model prediction.

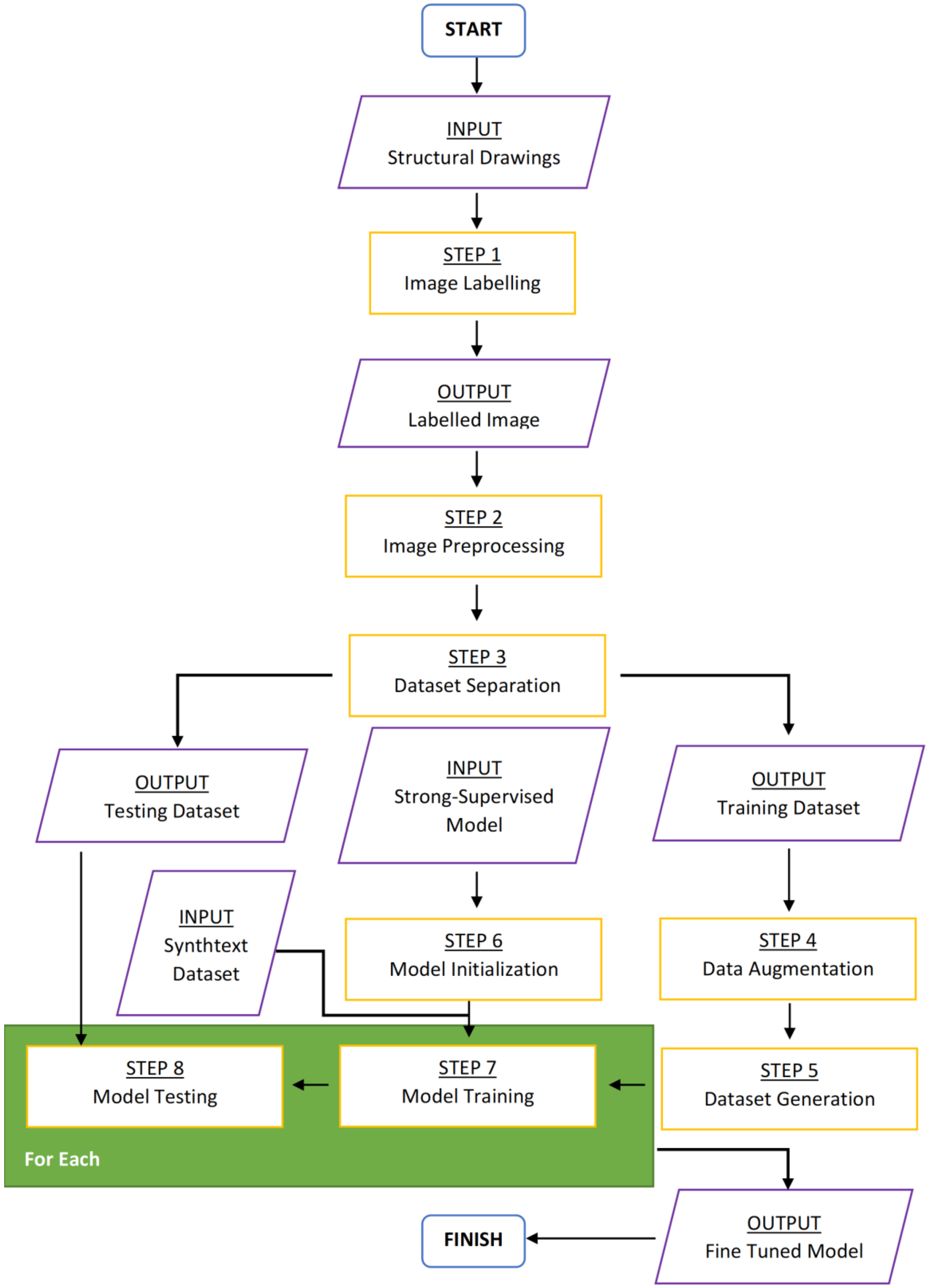


Figure 3.1. Research Workflow

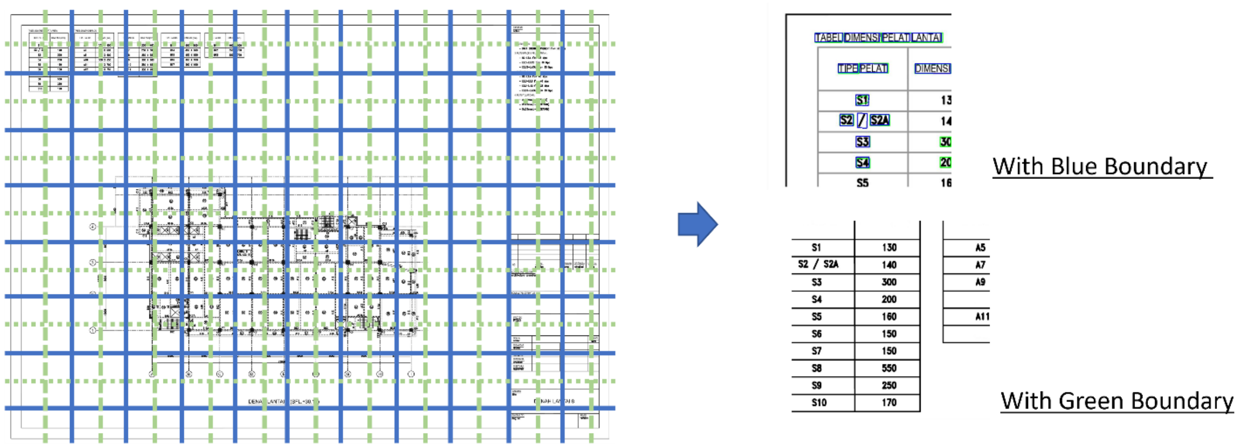


Figure 3.2. Image Cropping with Overlapping Boundary

3.3 Dataset Separation

In this step, the training and testing image datasets are separated. The number of training images is about 60 percent of the total set of images, while the remaining images are for testing purposes.

3.4 Image Rotation for Data Augmentation

The orientation of the texts inside the structural drawings is sometimes not aligned with either the horizontal or vertical direction. Therefore, additional rotated images are added for inclusion in the training dataset. This type of data augmentation is not applied in the testing dataset. Angles of 45 and 90 degrees, rotated in the counterclockwise and clockwise direction, are both implemented.

3.5 Dataset Generation

As explained in the previous chapter, text detection requires a character bounding box and an affinity bounding box. Using high-level annotation of the image, such as SynthText dataset, is important for the ground truth. However, it would be hard to do high-level annotation to all the labelled datasets. Therefore, the pseudo ground truth is generated using the watershed algorithm based on the cropped text, to ensure there is no disturbance, such as lines or fill pattern.

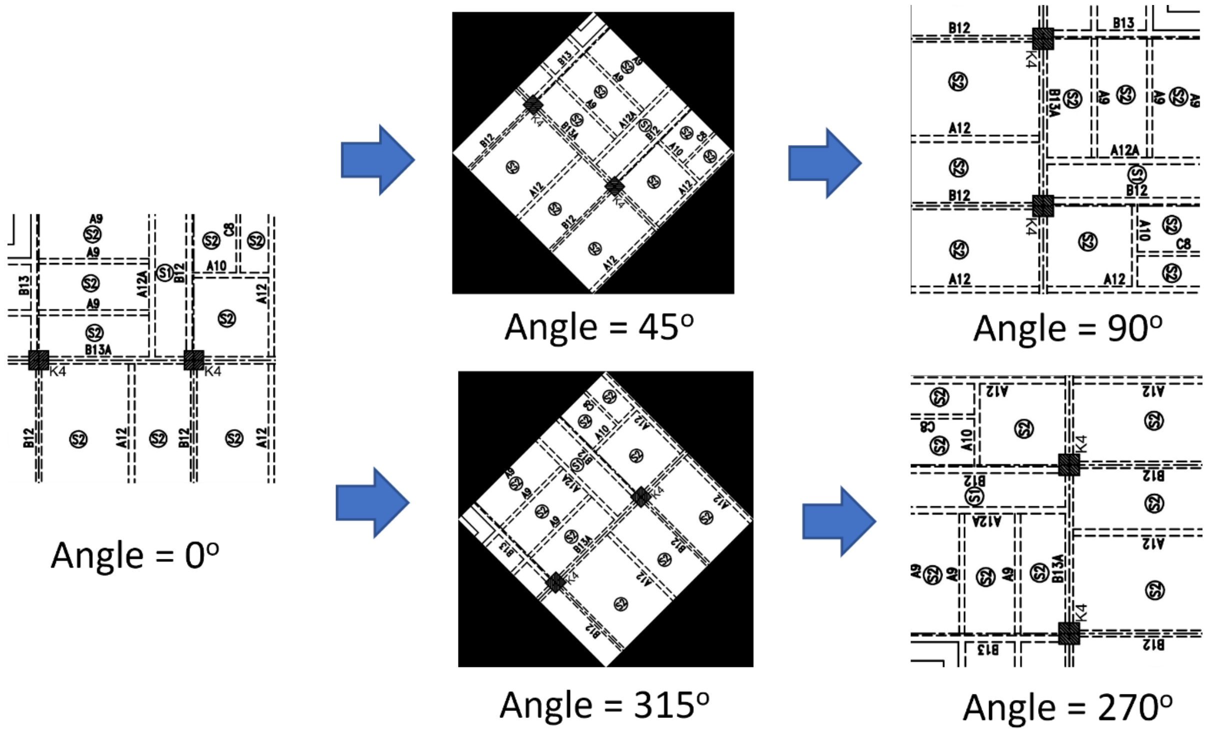


Figure 3.3. Image Rotation for Data Augmentation

3.6 Model Initialization

The previous strong supervision model with the SynthText data is loaded to the predictive system. This model would be trained and tested using the generated dataset. Learning rate and momentum must both be defined at the beginning.

3.7 Model Training

Weakly supervised training is commenced inside the model training. The predicted output is compared to the ground truth output and quantified with the loss function. The value of the loss function is processed further becomes the new gradient for the weight model training.

3.8 Model Testing

The performance of the model is tested using a different dataset. The F-Score, which consists of both prediction and recall performance, is monitored. This step is important to determine when to define the training completion. The performance of the model with the train dataset is also monitored to avoid overfitting in the training dataset. If the model performance based on the training dataset is increasing but the model performance based on the testing dataset is decreasing, the training is completed, and the highest performance of the predictive model is utilized.

4. NETWORK ARCHITECTURE

In this research, the CRAFT algorithm is adopted and explored for text detection. The implementation of the CRAFT fine tuning training programming code is based on works of Singh (2021). The input to the algorithm is an image containing text, and the network architecture consists of the image down sampling process, the image up sampling process, and the evaluation. The output to the algorithm is the image score, which is used for determining the level of the success of the code for the application here, reading structural drawings. Before inputting each of the images into the algorithm, the structural drawings need to be pre-processed. The image pre-processing steps was explained in Chapter 3.

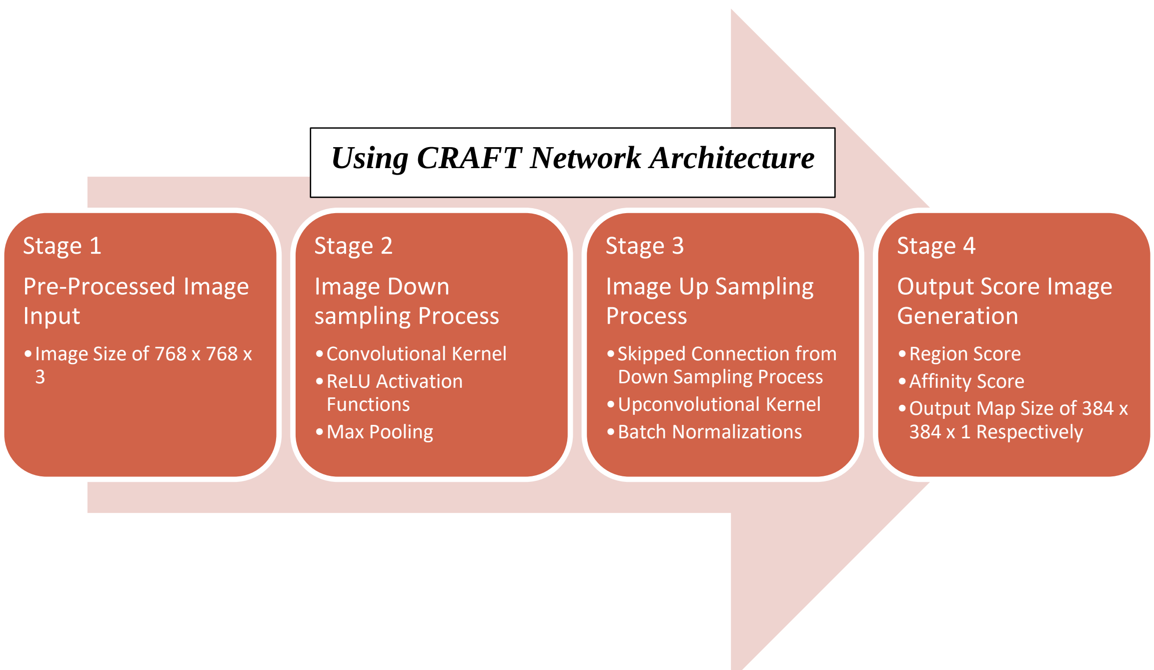


Figure 4.1. Workflow for Image Processing in Text Detection using the CRAFT Algorithm

In this study, the image input size of each image input to the algorithm is 768 x 768 x 3 pixels, which follows the strong supervision image input assumption. It is necessary to follow the strong supervision image size input assumption because the feature extraction kernel is already trained at this specific image size. If the dimension does not match this size, it will affect the training results.

The image input to the algorithm is processed further using a down-sampling process. It adopts VGG16-BN as the backbone of the feature extraction. This specific process is executed in six phases. At each phase, the image is first downsized to be half of the original width and height dimension, while increasing the number of channels by using 3 x 3 convolutional kernels and ReLU activation functions, and then applying max pooling to downsize the image dimension. After down-sampling, the extracted features are then up-sampled to the original dimension by 4 convolutional blocks. At each convolutional block, the input signal is processed by [1 x 1 x number of inputted channel] convolutional kernel, followed by a batch normalization. Then, the intermediate signal is processed further with [3 x 3 x number of inputted channel / 2] convolutional kernel, followed by the further batch normalization. In addition, there are skipped connections from the downsizing output signal to the up-sampling process. The output of phase 5 in down-sampling process is concatenated to the output of phase 6 in down sampling process. Then, the output of phase 4 in down-sampling process is concatenated to the output of first up-sampling convolutional block. The skipped connections also existed at the output of phase 3 in down sampling process is concatenated to the output of second up-sampling convolutional block, and the output of phase 2 in down-sampling process is concatenated to the output of third up-sampling convolutional block. After the 4 up-sampling convolutional blocks, the up-sampling process continued to the further up-sampling by convolutional kernels without batch normalizations. Then the output of the score can be obtained. There are two scores generated by the algorithm, which are region score and affinity score. These scores were defined in Chapter 3. Then, these scores are converted to the heatmap and post-processed to determine the text detection bounding boxes. The clear illustration of the network architecture is available in Figure 2.14.

The proposed architectural network has 20,770,466 parameters to be considered in predictive modelling.

5. DATASET AND FINE-TUNING RESULT

This chapter explains about the dataset description, the training parameters and the loss function used in fine-tuning training, and also the training results. As a matter of fact, different training parameters can yield different training results. So, the training parameters and training results needs to be stated clearly in this chapter to make the fine-tuning of the pretrained predictive model is reproducible.

5.1 Dataset Description

The dataset consists of computer-generated structural drawings which have different words shapes and type of fonts with the text being in various text orientations. The structural drawings are floor plans and structural details which are full of structural element labels and dimensions. The structural drawings available for this research are separated into two groups, including 6 sets of structural drawings for the training dataset and 5 sets of structural drawings for the testing dataset. Because the dimensions of the structural drawings are very large, the drawings are partitioned into smaller images with about 768 x 768 pixel to avoid their quality to be downgraded due to over resizing. To compensate for the possibility of cropped words when partitioning the images, a 50% overlapping partition is also used. After this pre-processing, the number of words in the training dataset and the testing dataset are 2942 words and 2014 words respectively.

The training dataset is augmented by 4 types of rotations to make the training more significant. The testing dataset is not augmented because it is rare to have extreme variations in orientation. Based on this pre-processing procedure, there are in total 3255 training images, including the rotated images and there are 387 testing images. The images in a training batch are shuffled randomly and SynthText data is still utilized with 16.66% probability to retain the strong supervision characteristic.

5.2 Training Parameters

The loss function that is utilized for the training with the word-level annotated sample needs to consider the pseudo ground truth confidence level. Let *w* be the sample of the training data, $R(w)$ and $l(w)$ be the region of the bounding box and the sample word length, respectively. From the character splitting process by the watershed algorithm, we can approximate the character bounding box and measure all the characters length $l^c(w)$. The confidence score $s_{conf}(w)$ can be calculated as follows:

$$s_{conf}(w) = \frac{l(w) - min(l(w), |l(w) - l^c(w)|)}{l(w)}$$

and the pixel-related confidence map S_c in the region of the bounding box $R(w)$ is computed as follows:

$$S_c(p) = \begin{cases} s_{conf}(w) \rightarrow p \in R(w) \\ 1 \rightarrow otherwise \end{cases}$$

where p is a pixel which should be inside the bounding box for the applied pixel-related confidence map. Then, the loss function L is defined as:

$$L = \sum_p S_c(p) \, [||S_r(p) - S_r^*(p)||_2^2 + ||S_a(p) - S_a^*(p)||_2^2]$$

where $S^*_r(p)$ and $S^*_a(p)$ are the pseudo-ground truth region score and affinity map which are generated by the watershed algorithm, and $S_r(p)$ and $S_a(p)$ are the predicted region score and affinity map. Furthermore, $S_c(p)$ is set to 1 to the SynthText data since the words are annotated to the character-level.

As for the training of the important hyperparameters, the batch size is set to be 4 and the number of iterations is 163 iterations at each epoch. The learning rate of each epoch is:

- Epoch 0 – 105 : 1 E -4
- Epoch 106 – 153 : 1 E -5
- Epoch 154 – 159 : 5 E -6

with no momentum method applied (the value is set to 1.0). The number of epochs is increased until the performance target is achieved, which occurs at about 159.

5.3 Fine Tuning Result

Based on the explained training method, defined hyperparameters, and pre-trained model, the fine tuning of the predictive model is commenced. The results of the training and testing performance at each epoch are displayed in Figure 5.1 to Figure 5.3 and summarized in

Table 5.1. The result shows that there are high fluctuations in performance during the training. However, the detection performance was better from one epoch to another epoch generally way.

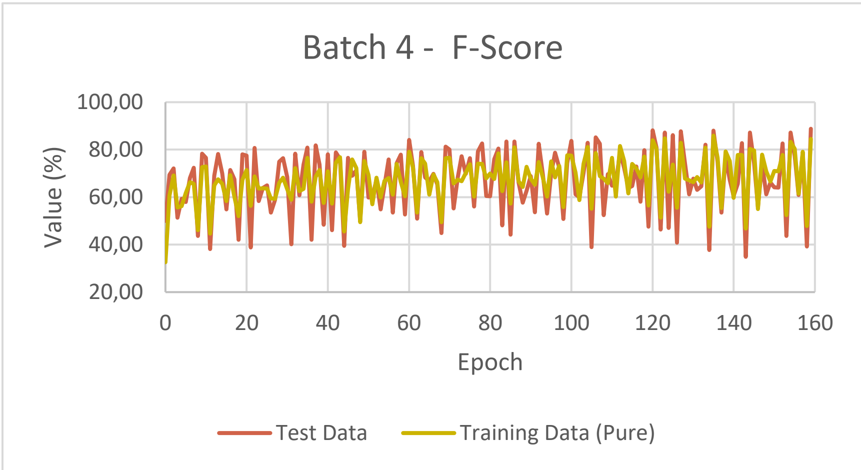


Figure 5.1. F-Score of Batch 4 Training Result

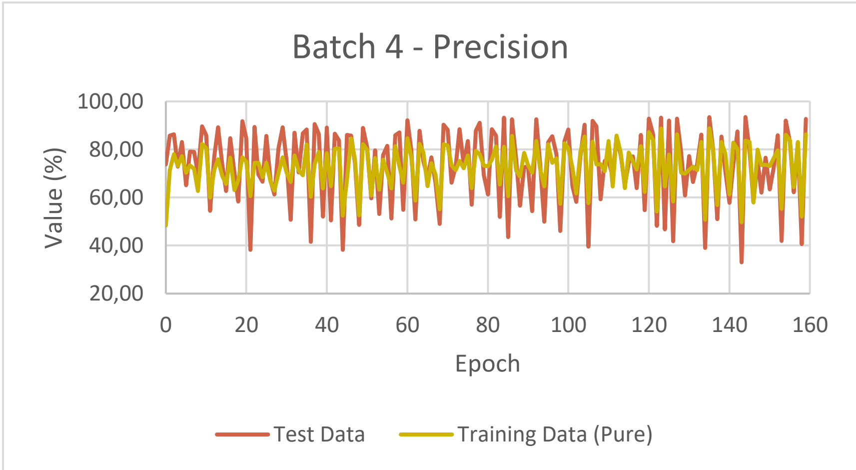


Figure 5.2. Precision of Batch 4 Training Result

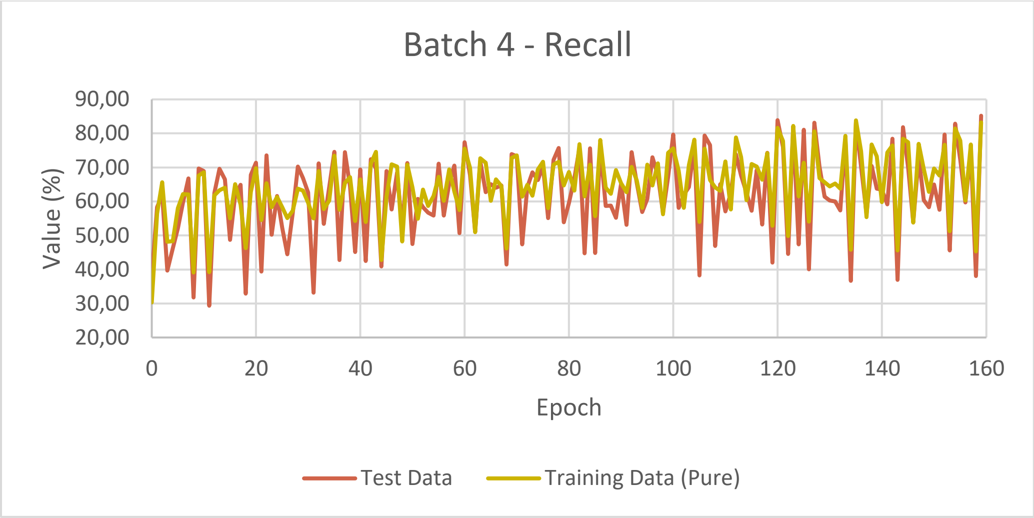


Figure 5.3. Recall of Batch 4 Training Result

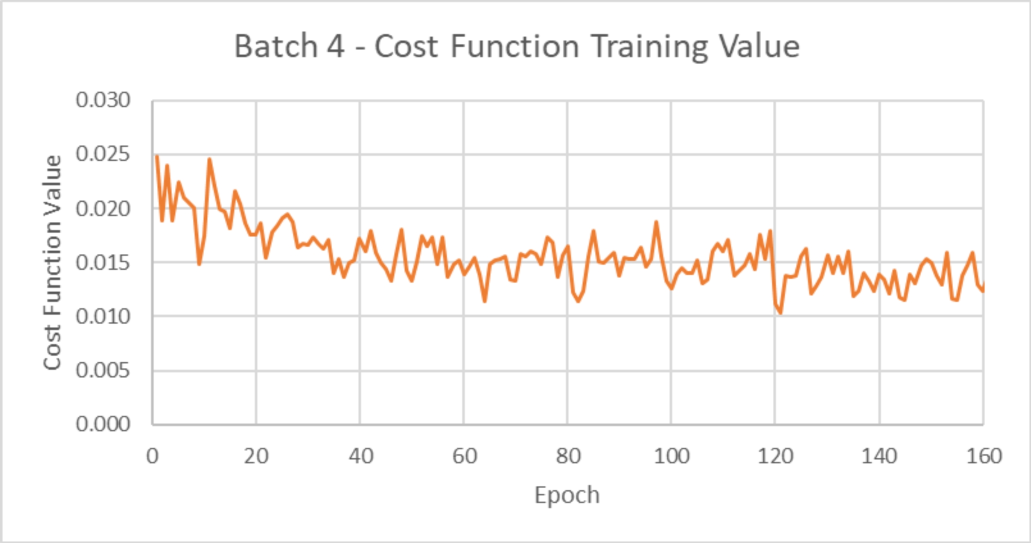


Figure 5.4. Cost Function Value of Batch 4 Training Result

Table 5.1. Training and Test Performance (Batch Size = 4)

Epoch	Training (Pure) - Initial Score				Test - Initial Score		
	F-Score	Precision	Recall	Cost Function	F-Score	Precision	Recall
0	32.58	48.32	30.31	-	50.01	73.74	37.83
1	60.65	71.36	56.38	0.0248	69.39	85.78	58.26
2	69.10	78.01	65.63	0.01891	72.05	86.27	61.85
3	55.61	73.13	48.12	0.02397	51.33	72.82	39.63
4	56.15	77.26	48.33	0.01887	59.32	83.09	46.13
5	61.74	70.27	58.02	0.0224	57.89	65.15	52.09
6	66.33	73.34	62.20	0.02096	68.05	79.03	59.74
7	65.57	71.90	62.05	0.02055	72.35	78.99	66.74
8	46.13	62.64	39.12	0.02002	43.67	69.98	31.74
9	72.68	82.26	67.62	0.01489	78.37	89.56	69.66
10	72.91	80.62	68.58	0.01745	76.52	85.92	68.97
11	44.65	59.90	39.17	0.02455	38.14	54.46	29.34
12	65.01	71.45	61.76	0.02208	68.96	76.94	62.48
13	67.54	75.92	63.28	0.01997	78.17	89.15	69.60
14	65.59	69.01	64.08	0.01974	69.91	73.72	66.48
15	58.34	65.66	55.00	0.01814	54.83	62.69	48.73
16	69.33	76.54	65.03	0.02157	71.48	84.63	61.86
17	60.12	63.09	59.02	0.02047	67.55	70.40	64.92
18	52.09	67.12	46.28	0.01866	42.09	58.26	32.94
19	67.29	76.72	62.85	0.01763	77.98	91.77	67.80
20	71.46	74.98	69.76	0.01757	77.45	84.62	71.40
21	56.32	60.54	54.53	0.01859	38.77	38.14	39.42
22	68.67	74.46	65.31	0.0155	80.70	89.41	73.54
23	63.57	74.39	58.28	0.01782	58.32	69.58	50.19
24	63.75	68.43	61.35	0.01842	63.95	66.53	61.56
25	63.61	74.45	58.27	0.01917	64.90	85.64	52.25
26	59.37	66.56	55.11	0.01947	53.52	67.12	44.50
27	59.27	62.85	57.30	0.01872	58.85	61.30	56.59
28	66.10	70.19	63.76	0.01637	75.00	80.50	70.21
29	68.23	76.72	63.22	0.01672	76.47	89.27	66.88
30	63.39	70.49	59.62	0.01662	68.49	75.51	62.67
31	58.96	66.48	55.02	0.01737	40.15	50.77	33.20
32	72.27	77.53	68.98	0.01669	78.25	86.92	71.16
33	62.52	71.20	57.70	0.01633	60.73	70.37	53.41
34	63.44	69.17	60.32	0.01706	73.77	86.68	64.21
35	76.60	82.07	73.41	0.01405	80.80	88.22	74.53
36	58.04	60.30	57.56	0.01529	42.11	41.49	42.75
37	68.63	73.86	65.47	0.01368	81.75	90.53	74.52
38	71.42	78.91	67.12	0.01493	73.77	86.35	64.39
39	57.58	63.79	54.30	0.01524	48.35	52.04	45.14
40	70.94	78.38	66.50	0.01725	77.99	89.08	69.36

Table 5.1. continued

Epoch	Training (Pure) - Initial Score				Test - Initial Score		
	F-Score	**Precision**	**Recall**	**Cost Function**	**F-Score**	**Precision**	**Recall**
41	57.33	64.75	54.00	0.01599	46.16	50.49	42.51
42	74.49	80.31	71.12	0.01794	78.82	86.51	72.38
43	76.76	80.48	74.57	0.01597	76.07	83.79	69.65
44	45.54	52.49	42.78	0.01501	39.50	38.19	40.90
45	63.54	67.38	61.44	0.01438	76.54	86.00	68.95
46	75.91	84.61	70.92	0.01332	68.96	85.79	57.65
47	72.23	75.94	70.29	0.01584	71.31	74.38	68.49
48	49.47	52.61	48.27	0.01805	50.32	48.62	52.14
49	75.19	82.16	70.85	0.01425	79.15	89.00	71.27
50	69.29	79.92	64.11	0.01326	59.84	81.00	47.44
51	56.97	60.85	54.94	0.01503	60.18	59.62	60.76
52	68.13	76.45	63.45	0.01747	67.28	79.60	58.26
53	59.83	63.26	58.71	0.01655	54.86	53.13	56.70
54	66.88	75.81	61.58	0.01735	65.00	77.69	55.87
55	68.04	70.26	67.26	0.01483	75.92	81.45	71.09
56	61.09	63.74	60.20	0.01729	53.47	51.30	55.84
57	73.98	81.30	69.42	0.01369	74.26	85.91	65.39
58	66.39	73.01	62.95	0.01484	77.94	87.10	70.53
59	60.41	66.14	57.41	0.01517	52.69	54.85	50.69
60	79.14	84.69	75.54	0.01396	84.10	92.11	77.36
61	72.46	76.88	69.91	0.01461	73.24	79.82	67.67
62	53.46	58.74	50.93	0.01542	50.97	50.87	51.08
63	76.35	82.26	72.72	0.01387	78.94	87.69	71.78
64	74.17	78.67	71.37	0.01136	68.35	75.09	62.72
65	61.12	64.66	60.16	0.01487	66.70	68.37	65.11
66	69.52	74.31	66.44	0.01516	69.78	76.66	64.04
67	66.32	69.55	64.49	0.01535	65.35	66.01	64.71
68	49.42	55.13	46.16	0.01559	44.91	49.01	41.45
69	76.44	82.11	72.83	0.01342	81.26	90.29	73.87
70	76.58	81.59	73.56	0.01327	80.02	88.19	73.23
71	65.82	73.38	61.39	0.01576	55.27	66.22	47.43
72	67.39	71.31	64.89	0.01561	68.33	72.95	64.26
73	66.75	75.26	61.71	0.01604	77.21	88.42	68.52
74	70.28	72.14	69.39	0.01576	69.85	73.81	66.29
75	73.82	77.47	71.64	0.01484	76.35	83.42	70.39
76	60.18	63.91	57.97	0.0173	56.05	57.00	55.13
77	74.07	79.32	70.78	0.01692	79.23	87.75	72.22
78	74.05	78.00	71.66	0.0137	82.65	91.01	75.69
79	67.87	73.25	64.74	0.01567	60.58	69.19	53.87
80	69.96	72.95	68.62	0.0165	60.40	61.29	59.53

Table 5.1. continued

Epoch	Training (Pure) - Initial Score				Test - Initial Score		
	F-Score	**Precision**	**Recall**	**Cost Function**	**F-Score**	**Precision**	**Recall**
81	67.67	75.79	63.24	0.01221	76.06	88.41	66.74
82	78.49	81.26	76.85	0.01146	80.40	85.74	75.69
83	62.56	65.40	61.50	0.01231	48.06	51.93	44.73
84	74.71	81.11	70.86	0.0156	83.48	93.15	75.63
85	57.33	60.63	55.71	0.01788	44.19	43.58	44.82
86	81.02	85.65	78.03	0.01506	83.46	92.54	76.00
87	66.77	71.26	64.07	0.01498	66.21	75.91	58.71
88	64.28	68.71	62.26	0.01549	57.64	56.55	58.76
89	72.82	78.53	69.25	0.01589	62.72	72.56	55.23
90	67.80	73.54	65.38	0.01373	68.84	72.92	65.19
91	65.28	70.32	62.74	0.01549	53.69	54.26	53.14
92	74.81	83.43	70.18	0.01536	82.54	92.53	74.50
93	68.17	71.74	66.18	0.01535	69.41	73.86	65.47
94	60.20	64.52	58.16	0.01639	53.15	49.91	56.85
95	75.08	82.13	70.82	0.01458	70.15	82.98	60.76
96	68.37	74.46	64.72	0.01533	78.72	85.49	72.94
97	73.15	76.36	71.15	0.01881	72.48	78.03	67.67
98	55.85	57.36	56.24	0.01557	50.83	46.12	56.61
99	77.64	82.55	74.30	0.01337	74.44	83.36	67.25
100	77.46	80.61	75.62	0.01265	83.72	88.27	79.61
101	70.30	72.49	69.07	0.01387	61.25	64.71	58.14
102	58.74	61.70	58.08	0.0145	60.02	58.20	61.95
103	74.05	79.14	70.74	0.014	70.19	77.39	64.21
104	81.10	85.36	78.18	0.01406	82.90	90.33	76.59
105	55.18	57.69	54.00	0.01526	38.92	39.58	38.28
106	78.57	83.25	75.52	0.01307	85.12	91.81	79.34
107	68.91	73.77	66.29	0.01338	82.55	89.62	76.51
108	67.81	73.86	64.14	0.01605	52.38	59.24	46.95
109	65.93	70.98	63.05	0.01669	69.74	75.19	65.03
110	76.50	83.52	71.74	0.01607	64.78	74.95	57.04
111	60.31	64.64	57.67	0.01711	65.50	68.92	62.40
112	81.52	85.72	78.77	0.01382	79.14	85.26	73.84
113	75.22	78.65	73.05	0.01431	72.25	77.62	67.57
114	61.57	63.95	60.38	0.01474	63.92	65.36	62.54
115	73.93	78.68	71.04	0.01585	64.74	74.39	57.30
116	72.06	75.03	70.22	0.01443	72.92	77.06	69.20
117	68.33	71.58	66.42	0.01756	58.13	63.97	53.26
118	77.09	81.34	74.23	0.01533	79.73	85.98	74.32
119	56.42	62.27	52.88	0.01792	47.51	54.65	42.03
120	83.91	87.26	81.65	0.01113	88.13	92.80	83.91

Table 5.1. continued

Epoch	Training (Pure) - Initial Score				Test - Initial Score		
	F-Score	**Precision**	**Recall**	**Cost Function**	**F-Score**	**Precision**	**Recall**
121	80.33	84.05	77.78	0.01032	80.87	86.37	76.03
122	51.41	54.21	49.85	0.01378	46.34	48.23	44.58
123	84.82	88.64	82.24	0.01363	87.16	93.14	81.90
124	62.26	64.54	61.46	0.01383	47.03	46.71	47.36
125	73.78	77.86	71.42	0.01556	86.17	91.96	81.06
126	55.50	58.29	54.03	0.01629	40.89	41.81	40.02
127	82.84	86.31	80.62	0.0121	87.70	92.74	83.18
128	67.74	70.34	66.90	0.01288	74.20	78.62	70.24
129	67.18	69.95	65.66	0.01366	61.13	60.85	61.41
130	66.35	71.51	64.41	0.01571	67.74	77.26	60.31
131	68.41	72.89	65.29	0.01405	63.03	66.42	59.97
132	66.11	71.36	63.80	0.01559	64.62	73.91	57.40
133	80.84	83.10	79.26	0.01404	82.14	86.18	78.47
134	47.59	50.72	45.82	0.016	37.80	38.96	36.70
135	85.91	88.82	83.84	0.01189	88.04	93.43	83.23
136	76.55	79.52	74.72	0.0124	74.73	79.84	70.24
137	55.45	56.87	55.38	0.014	53.53	50.95	56.38
138	79.22	83.10	76.78	0.01328	77.22	85.41	70.47
139	75.27	78.27	73.26	0.01231	67.25	71.13	63.78
140	59.68	61.20	59.77	0.0139	60.49	57.81	63.44
141	77.77	82.83	74.42	0.01339	65.61	73.68	59.13
142	77.78	80.11	76.48	0.01209	82.69	87.46	78.41
143	46.82	49.62	45.63	0.01428	34.88	33.01	36.96
144	80.37	83.62	78.41	0.01177	87.26	93.41	81.86
145	79.64	83.06	77.43	0.01155	74.39	79.35	70.02
146	55.04	57.90	53.74	0.01387	55.97	58.17	53.92
147	77.95	79.87	76.90	0.01308	74.15	75.38	72.96
148	70.84	73.48	69.53	0.01473	61.18	62.00	60.37
149	66.59	73.62	62.80	0.01532	66.22	76.60	58.31
150	70.96	73.06	69.70	0.01499	64.13	63.31	64.97
151	70.87	76.36	67.61	0.0138	64.01	72.15	57.52
152	77.73	79.57	76.62	0.0129	82.67	85.90	79.66
153	52.38	55.23	51.22	0.0159	43.71	41.95	45.63
154	83.32	86.14	81.49	0.01169	87.21	92.05	82.85
155	80.29	83.85	77.88	0.0115	77.35	83.25	72.23
156	61.51	64.25	60.30	0.01382	60.88	62.13	59.68
157	79.15	83.08	76.78	0.01476	78.43	82.55	74.69
158	47.83	52.02	45.29	0.01594	39.26	40.57	38.04
159	84.45	86.24	83.25	0.01292	88.80	92.72	85.19

A comparison between the testing results of pretrained model and the testing results of the fine-tuned model are shown at Figure 5.5. Based on the results, the method of fine tuning has been successfully completed and produced significantly better performance than the pre-trained model. It is also shown that the recall is increased significantly. However, there are some line symbols that are not detected as texts because it is harder to differentiate these lines from text in structural drawings that typically will have a huge number of lines.

Before Fine Tuning (Epoch 0) [Blue = Ground Truth; Green = Prediction]	After Fine Tuning (Epoch 159) [Blue = Ground Truth; Green = Prediction]

Figure 5.5. Comparison of Text Detection Result

Figure 5.5. Continued

Before Fine Tuning (Epoch 0) [Blue = Ground Truth; Green = Prediction]	After Fine Tuning (Epoch 159) [Blue = Ground Truth; Green = Prediction]

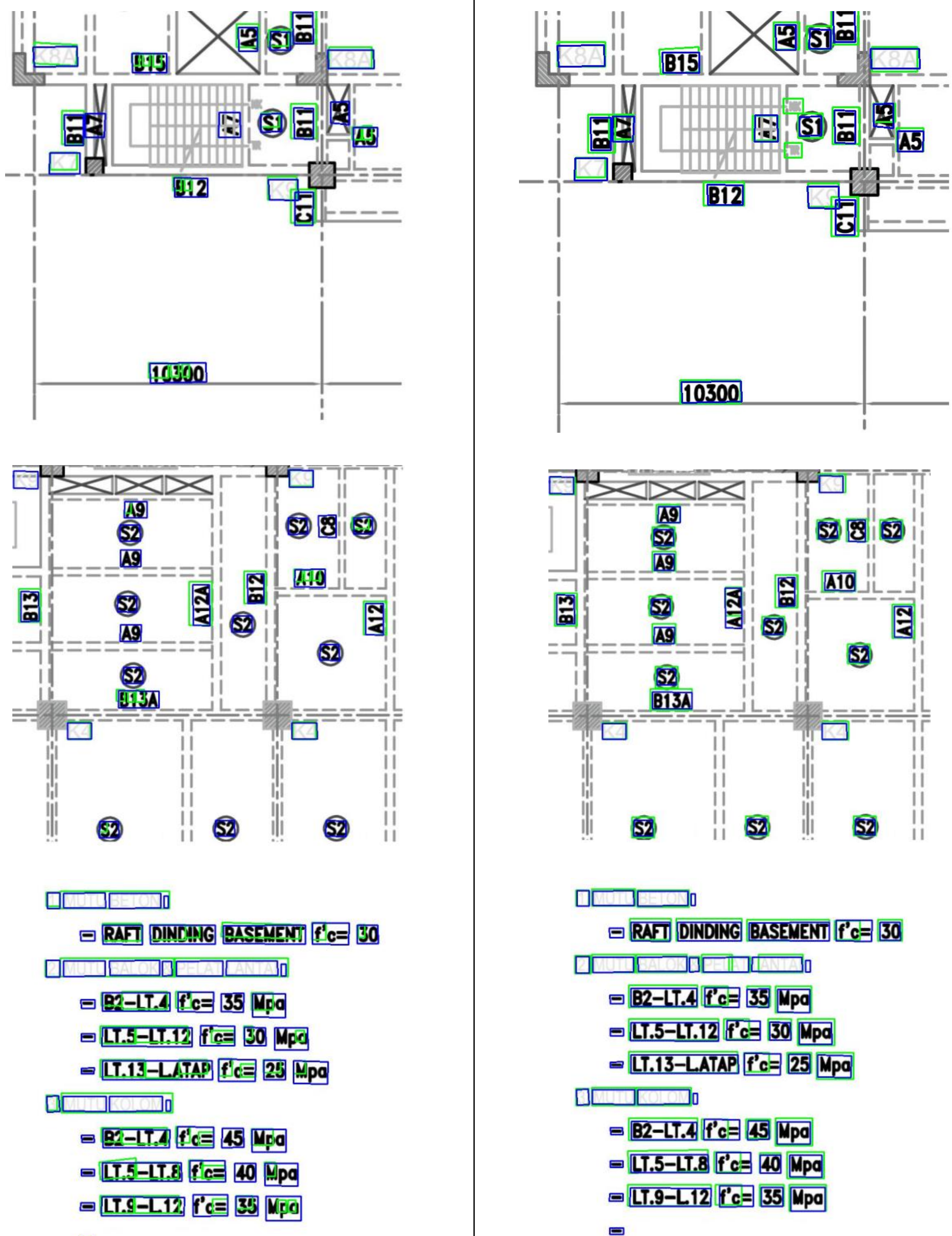
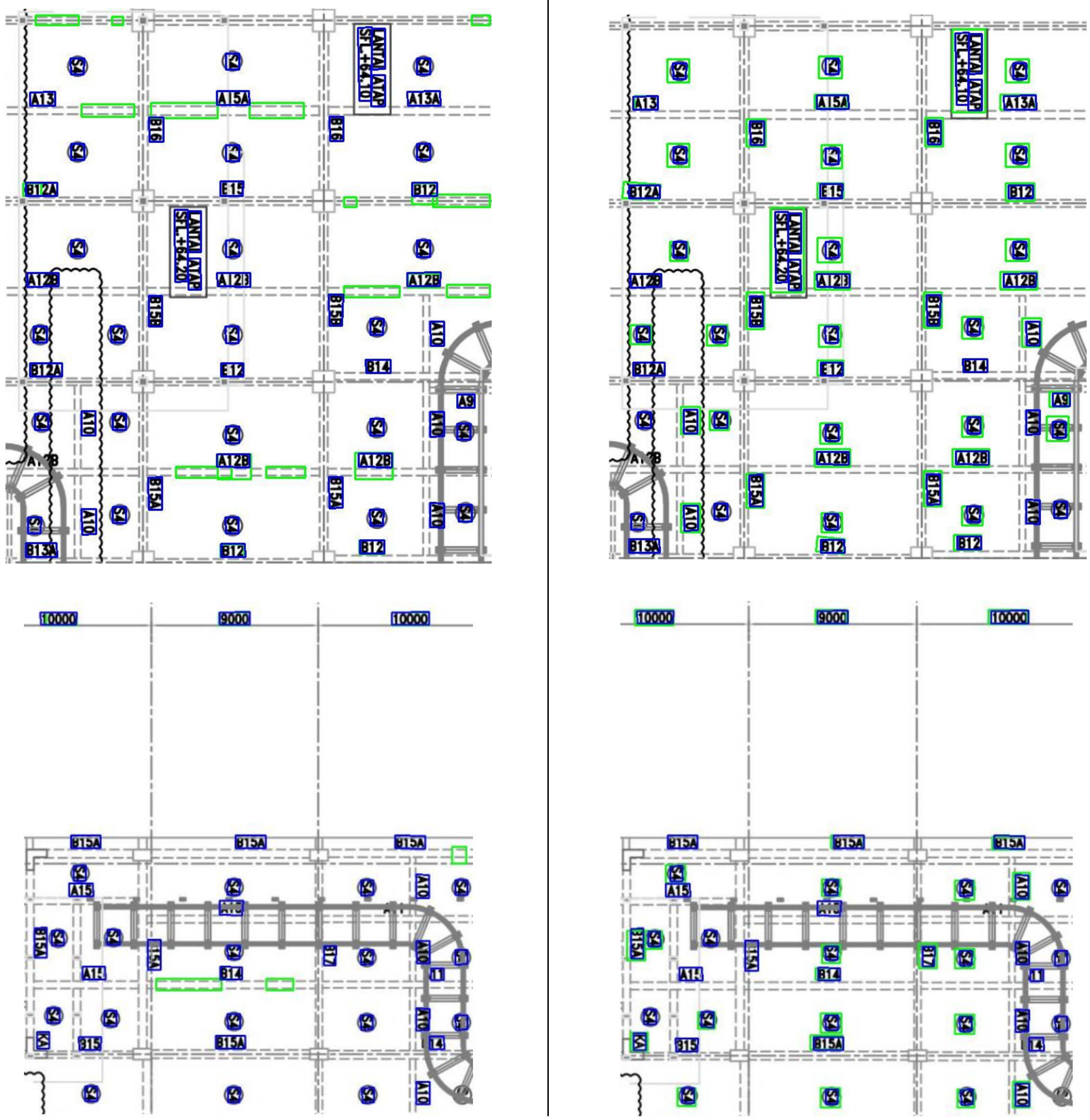


Figure 5.5. Continued

Before Fine Tuning (Epoch 0) [Blue = Ground Truth; Green = Prediction]	After Fine Tuning (Epoch 159) [Blue = Ground Truth; Green = Prediction]



5.4 Learned Lesson in Training

Acceptable performance in the predictive modelling was not obtained easily. There were several training failures beforehand that need to be highlighted and thus provide the good foundation for future work in the detection of text in structural drawings.

At first, the structural drawings were trained directly without any considerations given to the input image size. Since the training were limited to the GPU memory, we found that the image must be resized to a more appropriate dimension. This input dimension needs to be determined based on the structural drawings dimension since the structural drawings usually have a considerably larger dimension. The resizing from a higher dimension to a smaller dimension can affect the quality of the words' resolution, making it hard for the predictive model to learn and thus yielding unacceptable recall values. This issue can be easily solved by preprocessing the structural drawings through implementing a partitioning of the structural drawings into smaller parts.

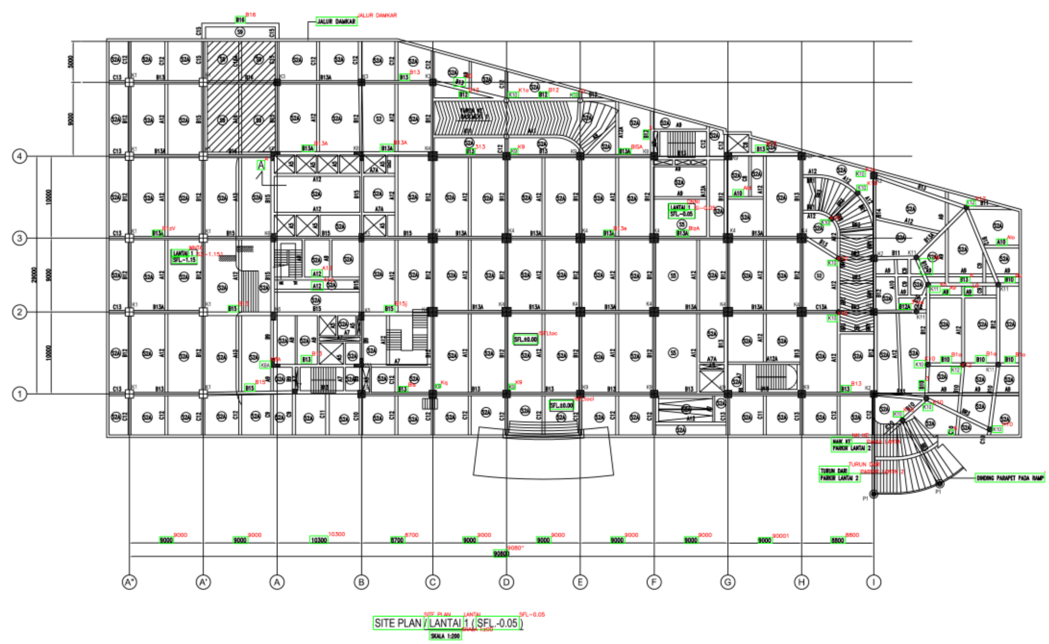


Figure 5.6. Output of Text Detection Using Original Dimension of Structural Drawing

Second, there were some difficulties for the initial model to learn the vertical oriented words and “noisy” surroundings due to the circle or lines nearby. This issue can be overcome by using data augmentation in the training by rotating the words by 45 degrees and 90 degrees, so the predictive model can learn to detect the words in various orientations.

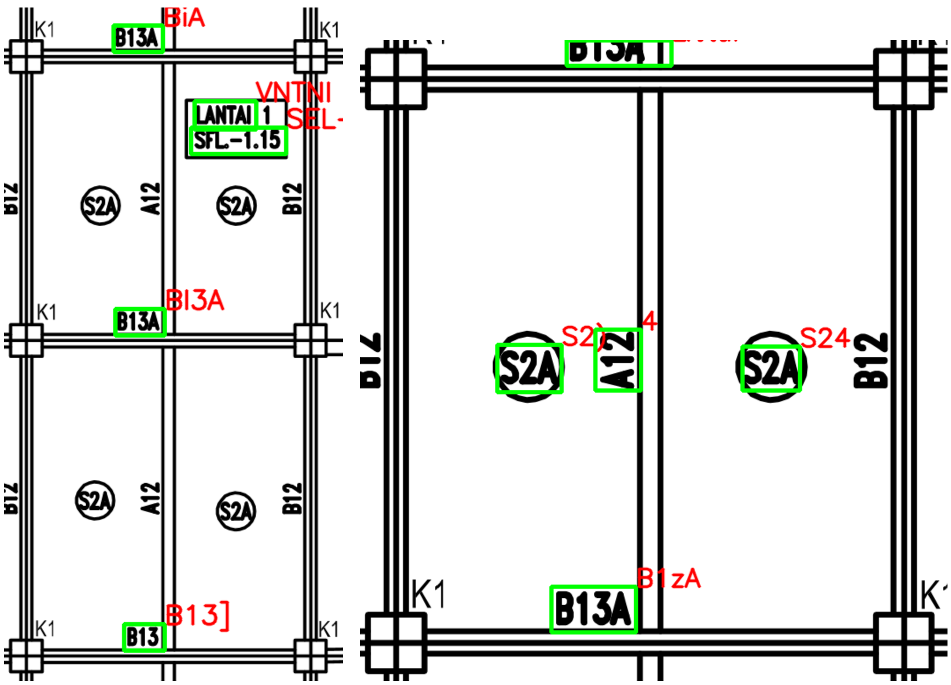


Figure 5.7. Sample of Initial Predictive Modelling

Third, it was hard to determine the appropriate training batch size and correct learning rate. In general trend, higher batch size yields more stable performance in training but requires more GPU memory. Its learning rate is usually larger compared to the smaller batch size. The determination of training batch size with the limited GPU memory was solved by iterating the training batch size values until the maximum batch size value was obtained and yielding no error in training process. Then, the learning rate was iterated until the training performance and testing performance were not fluctuating. Using higher batches does not guarantee higher performance of predictive modelling. The results of the training and testing performance with batch size of 6 and constant learning rate of 1E-6 at each epoch are displayed in Figure 5.8 to Figure 5.10 and summarized in

Table 5.2. The result shows that there were lower fluctuations in performance during the training but did not achieve better performance.

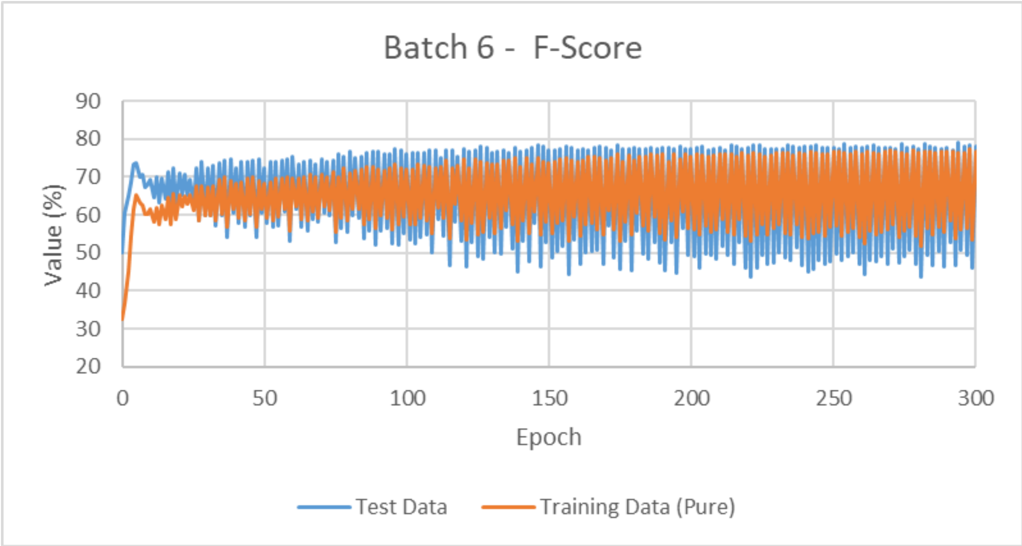


Figure 5.8. F-Score of Batch 6 Training Result

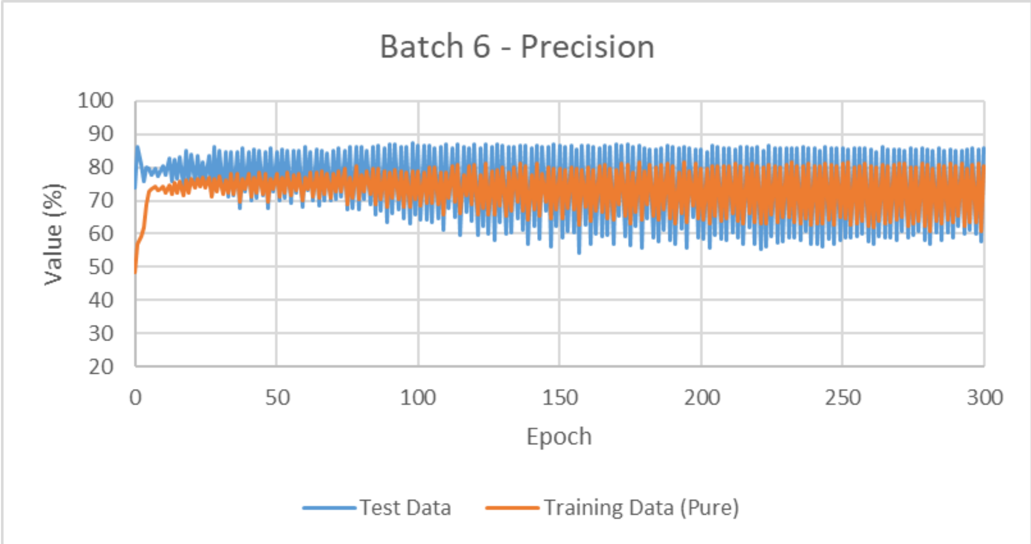


Figure 5.9. Precision of Batch 6 Training Result

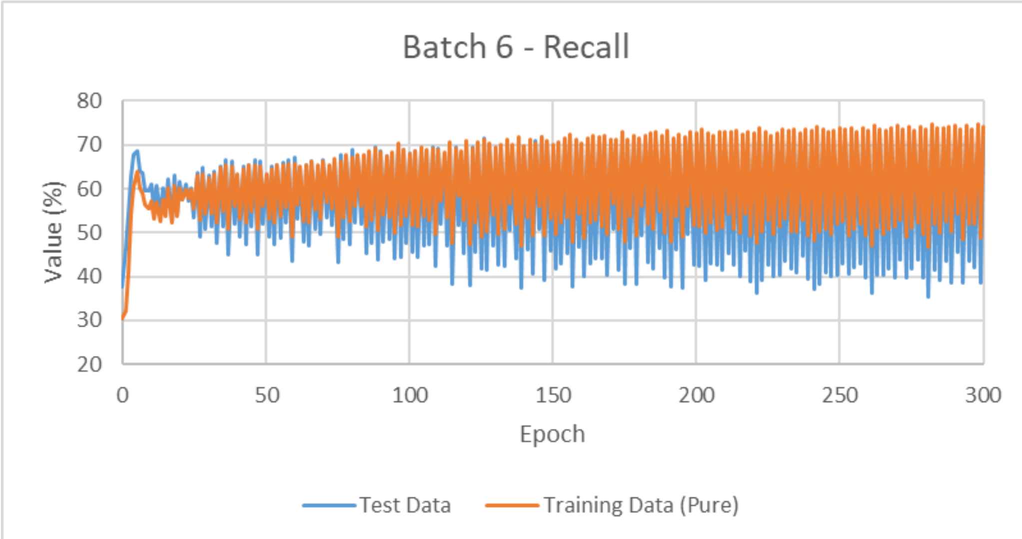


Figure 5.10. Recall of Batch 6 Training Result

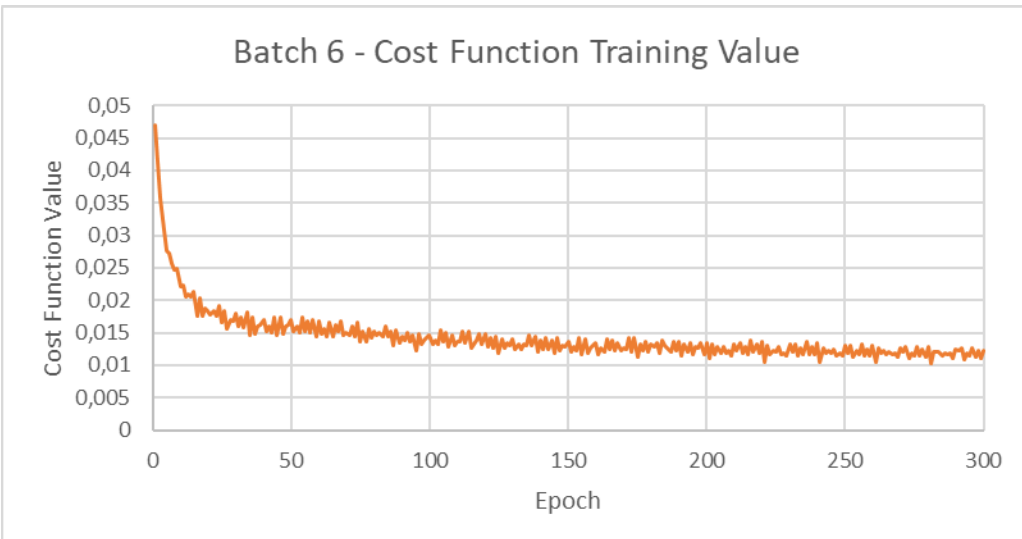


Figure 5.11. Precision of Batch 6 Training Result

Table 5.2. Training and Test Performance (Batch Size = 6)

Epoch	Training (Pure) - Initial Score				Test - Initial Score		
	F-Score	Precision	Recall	Cost Function	F-Score	Precision	Recall
0	32.59	48.32	30.32	-	50.01	73.74	37.83
1	36.54	56.94	32.14	0.04704	60.38	86.14	46.48
2	45.01	59.04	41.01	0.04122	64.88	80.70	54.24
3	54.56	61.94	53.95	0.03572	68.58	75.67	62.70
4	61.40	68.75	60.57	0.03071	73.31	80.04	67.62
5	65.35	72.62	63.98	0.02771	73.74	79.62	68.67
6	63.07	73.49	60.14	0.02723	70.07	77.61	63.86
7	62.67	74.16	59.01	0.02569	70.66	79.62	63.50
8	60.17	72.97	56.23	0.02467	67.36	77.27	59.69
9	60.05	73.38	55.56	0.02496	67.89	78.96	59.55
10	61.69	74.37	57.17	0.02206	69.40	80.57	60.95
11	58.06	72.35	53.21	0.02226	64.58	77.75	55.23
12	61.84	74.82	57.00	0.0206	70.06	82.63	60.80
13	57.51	71.82	52.57	0.02099	63.15	75.79	54.12
14	62.65	75.33	57.66	0.02053	69.58	82.46	60.18
15	58.76	72.46	53.66	0.02129	64.39	76.54	55.56
16	64.68	75.84	60.00	0.01758	71.18	83.04	62.28
17	57.60	71.68	52.29	0.02041	61.68	74.43	52.65
18	65.45	76.07	60.75	0.01761	72.42	85.11	63.02
19	58.93	72.24	53.62	0.01878	62.87	75.23	54.00
20	65.25	76.46	60.22	0.01831	71.03	83.95	61.56
21	62.39	73.71	57.48	0.01783	65.30	75.60	57.48
22	65.09	76.69	59.99	0.01844	70.50	83.59	60.95
23	62.87	74.37	57.88	0.01755	65.39	76.22	57.25
24	65.14	76.98	59.87	0.01916	69.21	81.61	60.08
25	61.09	74.01	55.42	0.01662	62.96	76.32	53.59
26	67.68	77.07	63.11	0.01835	72.27	83.41	63.76
27	58.45	71.16	52.78	0.0156	58.35	71.68	49.20
28	67.67	76.60	63.29	0.01695	73.96	86.10	64.82
29	60.01	72.86	54.32	0.01669	59.97	72.99	50.88
30	67.41	77.32	62.51	0.01788	72.44	85.08	63.07
31	59.91	72.09	54.34	0.01593	59.98	72.25	51.27
32	67.64	76.36	63.22	0.0173	72.90	84.75	63.95
33	58.47	72.13	52.34	0.01578	57.10	71.00	47.75
34	69.21	78.00	64.80	0.01826	73.56	84.58	65.08
35	60.29	72.13	54.78	0.01459	59.83	71.41	51.48
36	69.84	78.19	65.52	0.01733	74.46	84.57	66.51
37	56.88	69.75	50.80	0.0149	54.10	67.55	45.11
38	68.95	76.71	64.99	0.01592	74.79	85.97	66.19
39	61.48	73.42	55.93	0.01611	60.50	72.57	51.88
40	68.24	78.06	63.23	0.01697	72.22	84.64	62.97

Table 5.2. Continued

Epoch	Training (Pure) - Initial Score				Test - Initial Score		
	F-Score	Precision	Recall	Cost Function	F-Score	Precision	Recall
41	59.03	71.36	53.14	0.01529	57.78	70.17	49.12
42	68.67	76.83	64.41	0.01597	73.94	85.42	65.18
43	58.53	72.28	52.25	0.01509	56.70	70.68	47.33
44	69.81	78.45	65.31	0.01738	73.92	85.04	65.37
45	60.29	72.13	54.78	0.01459	59.83	71.41	51.48
46	69.84	78.19	65.52	0.01733	74.46	84.57	66.51
47	56.88	69.75	50.80	0.0149	54.10	67.55	45.11
48	68.95	76.71	64.99	0.01592	74.79	85.97	66.19
49	61.48	73.42	55.93	0.01611	60.50	72.57	51.88
50	68.24	78.06	63.23	0.01697	72.22	84.64	62.97
51	59.03	71.36	53.14	0.01529	57.78	70.17	49.12
52	68.67	76.83	64.41	0.01597	73.94	85.42	65.18
53	58.53	72.28	52.25	0.01509	56.70	70.68	47.33
54	69.81	78.45	65.31	0.01738	73.92	85.04	65.37
55	59.39	71.36	53.64	0.01515	57.32	69.28	48.87
56	69.64	77.84	65.26	0.01672	74.35	84.91	66.13
57	61.44	72.94	55.83	0.01493	60.75	72.26	52.40
58	70.00	78.04	65.69	0.0169	74.69	84.88	66.69
59	55.70	69.88	48.94	0.01438	53.10	68.04	43.54
60	69.59	76.92	65.65	0.01677	75.53	86.19	67.22
61	62.08	73.53	56.46	0.01482	61.46	73.09	53.02
62	69.76	78.47	65.08	0.01569	73.38	84.67	64.74
63	58.78	71.20	52.55	0.01445	56.85	70.16	47.78
64	70.01	78.30	65.54	0.01668	74.02	85.46	65.29
65	58.37	70.79	52.27	0.01446	55.86	68.64	47.09
66	70.68	78.82	66.22	0.01619	74.49	85.20	66.17
67	60.92	72.03	55.22	0.0153	58.81	70.04	50.68
68	69.91	78.27	65.40	0.01671	73.31	84.45	64.77
69	59.20	70.83	53.16	0.01469	58.19	70.19	49.69
70	70.55	78.03	66.36	0.01492	74.61	84.86	66.58
71	62.18	72.53	56.88	0.01468	60.68	70.27	53.39
72	70.09	79.01	65.29	0.01606	74.14	85.88	65.23
73	61.30	72.71	55.62	0.01436	59.84	71.17	51.62
74	71.42	79.76	66.84	0.01658	74.24	85.09	65.85
75	55.57	68.86	49.04	0.01369	52.69	67.44	43.23
76	70.09	76.80	66.38	0.01523	75.92	86.37	67.73
77	59.17	70.92	53.31	0.01361	56.50	67.85	48.41
78	72.39	80.54	67.86	0.01553	75.48	86.37	67.03
79	58.56	70.40	52.43	0.01415	55.59	67.30	47.35
80	71.04	77.82	67.26	0.01528	76.65	86.37	68.89

Table 5.2. Continued

Epoch	Training (Pure) - Initial Score				Test - Initial Score		
	F-Score	Precision	Recall	Cost Function	F-Score	Precision	Recall
81	61.60	72.53	56.00	0.01454	59.88	70.21	52.20
82	71.77	78.89	67.72	0.01503	75.21	85.21	67.32
83	61.63	71.91	56.16	0.0144	59.28	69.04	51.95
84	72.06	79.56	67.70	0.01594	75.48	86.13	67.17
85	57.35	69.01	51.34	0.01438	53.66	65.77	45.32
86	72.54	79.60	68.57	0.01525	76.36	86.68	68.23
87	58.85	70.27	52.88	0.01297	55.69	66.92	47.68
88	72.83	78.96	69.32	0.01548	76.67	85.67	69.39
89	56.53	67.91	50.59	0.01333	51.94	63.53	43.92
90	72.01	78.36	68.39	0.01435	76.80	86.98	68.75
91	59.81	70.92	54.03	0.01389	55.67	66.22	48.02
92	71.83	79.25	67.54	0.01501	75.99	86.88	67.52
93	59.35	70.39	53.53	0.01366	56.46	67.32	48.62
94	72.25	78.90	68.44	0.0147	76.07	86.29	68.01
95	56.25	68.27	50.07	0.01233	52.39	64.60	44.07
96	73.27	78.68	70.21	0.01476	77.35	86.12	70.21
97	57.49	68.19	51.82	0.01323	52.11	62.92	44.47
98	72.47	78.82	68.83	0.01389	76.91	87.18	68.79
99	59.54	70.47	53.81	0.01417	55.21	65.96	47.48
100	72.04	78.86	68.06	0.01463	76.09	86.49	67.93
101	57.58	68.65	51.75	0.01314	53.33	64.24	45.59
102	72.12	78.34	68.54	0.01386	76.50	86.57	68.52
103	57.40	69.06	51.39	0.01317	52.55	64.35	44.41
104	73.44	80.14	69.56	0.01538	76.54	86.54	68.62
105	59.16	69.31	53.62	0.01353	54.04	63.62	46.96
106	73.06	79.89	69.05	0.015	76.26	86.01	68.49
107	58.39	68.72	52.80	0.01283	54.64	64.69	47.30
108	72.65	78.55	69.23	0.01461	77.01	86.39	69.47
109	55.26	65.84	49.56	0.01298	50.02	61.05	42.36
110	72.45	78.32	69.03	0.01358	77.03	87.02	69.10
111	61.91	70.60	57.10	0.01362	58.79	67.57	52.03
112	72.69	80.44	68.18	0.01525	74.90	85.82	66.45
113	59.09	69.46	53.48	0.01338	54.55	65.06	46.96
114	74.38	80.83	70.60	0.01525	77.18	87.07	69.31
115	53.93	66.06	47.63	0.01268	46.54	59.69	38.14
116	72.47	77.88	69.24	0.01315	76.98	86.25	69.50
117	63.01	71.41	58.30	0.01385	58.25	66.49	51.83
118	73.05	80.59	68.57	0.01489	75.43	85.89	67.23
119	58.65	69.04	52.97	0.01314	53.09	63.96	45.39
120	74.62	80.65	71.04	0.0148	77.42	86.96	69.77

Table 5.2. Continued

Epoch	Training (Pure) - Initial Score				Test - Initial Score		
	F-Score	Precision	Recall	Cost Function	F-Score	Precision	Recall
121	53.53	65.62	47.25	0.01279	46.48	59.68	38.05
122	72.45	77.72	69.27	0.01418	76.78	85.92	69.39
123	57.72	67.62	52.29	0.01259	52.66	62.32	45.59
124	74.69	81.38	70.70	0.01442	77.04	87.04	69.10
125	54.86	65.58	48.99	0.0119	49.18	60.00	41.67
126	74.13	78.83	71.32	0.01403	78.16	86.15	71.53
127	55.64	65.63	50.13	0.01265	48.34	58.03	41.43
128	73.57	79.19	70.27	0.0135	77.62	87.00	70.06
129	59.20	68.45	54.00	0.01311	54.00	63.57	46.93
130	73.40	80.02	69.40	0.01404	76.29	86.60	68.17
131	55.20	65.21	49.64	0.01246	49.96	60.12	42.73
132	73.22	78.63	70.00	0.0133	76.68	86.07	69.13
133	56.36	66.77	50.70	0.01243	49.73	60.36	42.28
134	74.50	80.06	71.16	0.0132	77.00	86.29	69.52
135	61.30	69.57	56.59	0.01296	56.67	64.67	50.43
136	74.32	81.16	70.16	0.01465	76.29	86.46	68.26
137	57.34	66.94	51.95	0.01294	51.16	61.04	44.04
138	75.05	80.44	71.83	0.01447	77.61	86.61	70.31
139	53.13	64.31	47.15	0.0121	45.01	56.74	37.30
140	72.84	77.90	69.77	0.0138	76.85	85.85	69.55
141	58.43	67.32	53.39	0.01248	53.02	62.31	46.14
142	74.88	81.07	71.16	0.01413	77.19	86.71	69.55
143	55.18	65.32	49.48	0.01253	47.81	58.37	40.48
144	73.27	78.30	70.26	0.01384	77.82	86.22	70.92
145	60.91	69.45	56.07	0.01189	57.71	66.78	50.80
146	74.99	80.04	71.93	0.01413	78.29	86.14	71.75
147	54.82	64.51	49.38	0.01217	46.22	56.18	39.26
148	74.13	79.51	70.89	0.01297	77.95	87.05	70.58
149	58.33	67.39	53.19	0.01285	52.82	62.34	45.82
150	73.76	79.71	70.13	0.01359	77.02	86.69	69.29
151	55.02	64.65	49.67	0.012	48.78	58.66	41.75
152	73.95	79.13	70.79	0.01283	76.94	86.18	69.49
153	57.76	67.51	52.36	0.01226	50.27	60.67	42.91
154	74.88	80.10	71.65	0.01414	77.36	86.43	70.02
155	58.51	67.21	53.47	0.01156	51.72	60.24	45.31
156	75.15	79.79	72.35	0.01394	77.93	85.39	71.67
157	53.19	62.67	47.91	0.01191	44.36	54.07	37.60
158	74.28	79.25	71.24	0.01257	77.91	86.59	70.80
159	59.42	68.18	54.40	0.01304	53.46	62.50	46.70
160	74.09	80.19	70.33	0.01351	77.05	86.54	69.44

Table 5.2. Continued

Epoch	Training (Pure) - Initial Score				Test - Initial Score		
	F-Score	Precision	Recall	Cost Function	F-Score	Precision	Recall
161	54.32	63.96	48.85	0.01172	47.03	57.01	40.03
162	74.39	79.17	71.46	0.01268	77.46	86.24	70.31
163	57.47	66.78	52.30	0.01209	49.94	59.95	42.80
164	75.30	80.52	72.04	0.01405	77.04	85.93	69.82
165	57.94	66.43	52.99	0.0124	50.55	59.22	44.10
166	75.25	80.78	71.82	0.01387	77.63	86.52	70.40
167	56.24	64.72	51.32	0.01231	50.75	59.52	44.23
168	75.01	79.65	72.15	0.01336	78.08	86.34	71.25
169	54.93	64.05	49.78	0.01225	47.06	56.74	40.21
170	74.31	79.35	71.23	0.0127	77.85	86.82	70.56
171	61.18	68.78	56.78	0.01246	57.28	65.57	50.85
172	75.04	81.16	71.22	0.01425	76.92	86.61	69.18
173	56.10	65.53	50.76	0.01197	48.87	59.21	41.61
174	76.02	80.90	72.95	0.01429	78.35	86.82	71.38
175	53.28	63.07	47.78	0.01133	45.56	56.47	38.18
176	73.94	78.45	71.15	0.01321	77.49	86.21	70.37
177	58.34	66.41	53.62	0.01226	53.08	62.19	46.30
178	75.73	81.52	72.07	0.01354	77.54	86.59	70.19
179	54.88	64.16	49.49	0.01206	45.45	55.83	38.33
180	74.28	78.77	71.49	0.01322	77.83	85.77	71.24
181	60.62	68.67	55.95	0.01281	55.94	64.76	49.23
182	75.49	80.62	72.25	0.0135	77.29	85.41	70.58
183	57.02	65.28	52.11	0.01186	49.90	58.92	43.28
184	75.39	79.86	72.58	0.01385	77.36	85.55	70.61
185	55.83	64.27	50.93	0.01294	48.47	57.82	41.72
186	76.05	80.85	73.04	0.01255	77.75	85.79	71.08
187	59.55	67.22	55.00	0.01211	52.85	61.17	46.53
188	75.64	81.09	72.15	0.01367	77.41	86.53	70.03
189	55.17	64.77	49.81	0.01222	47.30	58.17	39.86
190	76.05	80.57	73.15	0.01357	77.86	86.18	71.00
191	53.41	63.25	47.86	0.0114	45.25	56.77	37.62
192	74.19	78.43	71.51	0.01318	77.66	85.89	70.87
193	58.35	66.17	53.77	0.01205	52.78	61.64	46.14
194	75.96	81.51	72.40	0.01341	77.48	86.24	70.34
195	54.27	63.46	48.92	0.01171	44.80	55.64	37.49
196	74.60	78.84	71.91	0.01291	78.00	85.68	71.58
197	60.76	68.54	56.18	0.01266	56.38	65.12	49.71
198	76.06	81.04	72.84	0.01337	77.45	85.41	70.85
199	56.41	64.49	51.55	0.01161	49.39	58.69	42.64
200	75.45	79.61	72.79	0.01347	77.40	85.45	70.74

Table 5.2. Continued

Epoch	Training (Pure) - Initial Score				Test - Initial Score		
	F-Score	Precision	Recall	Cost Function	F-Score	Precision	Recall
201	56.12	64.11	51.43	0.0111	49.03	58.27	42.32
202	75.98	79.75	73.60	0.01321	77.87	84.79	71.99
203	54.65	62.89	49.82	0.0117	46.03	55.59	39.28
204	75.76	80.61	72.71	0.01281	78.00	86.40	71.09
205	57.14	65.56	52.20	0.01183	49.85	59.46	42.91
206	75.32	80.50	71.98	0.01284	77.18	86.06	69.97
207	55.51	63.92	50.64	0.01181	49.31	58.52	42.60
208	75.78	80.34	72.86	0.01222	77.53	85.94	70.63
209	56.09	64.63	51.16	0.01142	48.49	58.19	41.56
210	75.93	80.29	73.13	0.01332	77.62	85.81	70.85
211	58.69	66.24	54.17	0.01227	51.61	60.21	45.16
212	76.18	81.20	72.87	0.01334	77.12	85.36	70.32
213	55.62	63.73	50.77	0.0118	49.34	58.75	42.52
214	76.02	80.30	73.25	0.01277	78.40	86.18	71.91
215	54.81	63.18	49.92	0.01171	46.95	56.79	40.02
216	75.27	79.87	72.33	0.0138	77.94	86.33	71.05
217	58.84	66.30	54.36	0.01203	52.74	61.79	46.00
218	76.23	81.26	72.94	0.01317	77.50	85.82	70.64
219	54.47	63.47	49.22	0.01177	46.11	56.77	38.83
220	75.73	80.45	72.67	0.01367	77.75	86.10	70.88
221	53.10	62.89	47.60	0.0105	43.77	55.49	36.14
222	76.24	80.01	73.87	0.01294	78.26	84.93	72.56
223	54.80	62.93	50.09	0.01171	46.08	55.97	39.16
224	76.05	80.63	73.09	0.01226	78.33	86.50	71.58
225	57.57	65.90	52.72	0.01208	49.54	59.26	42.56
226	75.52	80.59	72.23	0.01274	77.35	85.86	70.37
227	54.72	63.31	49.75	0.01156	47.18	57.25	40.13
228	75.66	80.13	72.76	0.01189	77.54	85.82	70.72
229	56.37	65.05	51.36	0.01149	47.54	57.70	40.42
230	76.41	80.90	73.46	0.01316	77.54	85.67	70.82
231	57.61	65.03	53.11	0.01201	50.09	58.85	43.60
232	76.50	81.46	73.22	0.01315	77.46	85.63	70.71
233	55.14	63.30	50.25	0.01147	48.84	58.69	41.82
234	76.29	80.44	73.57	0.0126	78.36	85.94	72.01
235	54.93	62.99	50.13	0.0117	48.02	58.06	40.93
236	75.77	80.49	72.75	0.01353	78.14	86.35	71.37
237	57.36	64.73	52.91	0.01172	51.53	60.74	44.74
238	76.56	81.30	73.44	0.01278	77.80	85.97	71.05
239	54.29	62.73	49.30	0.0116	46.61	56.92	39.45
240	76.09	80.47	73.22	0.01342	77.94	85.95	71.30

Table 5.2. Continued

Epoch	Training (Pure) - Initial Score				Test - Initial Score		
	F-Score	Precision	Recall	Cost Function	F-Score	Precision	Recall
241	53.65	62.98	48.29	0.01046	44.98	57.01	37.14
242	76.59	80.31	74.20	0.01259	78.24	84.91	72.54
243	55.08	63.24	50.34	0.01175	45.56	56.09	38.36
244	76.50	81.01	73.60	0.01222	78.50	86.24	72.04
245	55.64	64.05	50.73	0.01178	48.04	58.29	40.85
246	76.17	80.89	73.08	0.01246	77.60	85.73	70.88
247	54.48	62.79	49.62	0.01145	46.99	56.96	39.98
248	76.16	80.41	73.35	0.0118	77.79	85.69	71.22
249	56.73	65.10	51.84	0.01153	47.57	58.06	40.29
250	76.69	81.06	73.81	0.01296	77.64	85.66	71.00
251	57.02	64.51	52.46	0.0119	49.74	58.93	43.04
252	76.78	81.42	73.69	0.01294	77.56	85.54	70.95
253	54.31	62.69	49.29	0.01119	48.15	58.89	40.72
254	76.40	80.17	73.91	0.0122	78.75	85.89	72.70
255	55.52	63.06	50.93	0.0116	49.13	59.33	41.93
256	76.02	80.52	73.12	0.01321	77.80	85.78	71.17
257	56.42	63.87	51.88	0.01153	50.15	60.03	43.07
258	76.83	81.11	74.00	0.01251	78.21	85.94	71.75
259	54.23	62.37	49.35	0.01146	46.97	57.60	39.65
260	76.10	80.58	73.21	0.013	77.78	85.52	71.32
261	52.52	61.76	47.16	0.01045	44.22	56.94	36.14
262	76.73	80.14	74.52	0.01254	78.25	84.63	72.77
263	55.82	63.78	51.11	0.0119	47.96	59.10	40.35
264	76.40	80.97	73.43	0.01216	78.28	86.17	71.72
265	54.67	63.04	49.67	0.01158	47.80	58.98	40.18
266	76.14	80.31	73.40	0.01198	77.87	85.52	71.48
267	55.49	63.48	50.77	0.01169	48.94	58.93	41.85
268	76.61	80.71	73.89	0.01184	77.62	85.35	71.17
269	56.07	64.04	51.33	0.01119	47.09	58.01	39.63
270	77.05	81.08	74.38	0.01263	77.74	85.45	71.30
271	57.53	65.04	52.94	0.01195	50.93	60.78	43.83
272	76.71	81.18	73.71	0.01283	77.49	85.24	71.03
273	54.09	62.41	49.02	0.01114	47.39	58.72	39.73
274	76.55	80.06	74.23	0.01194	78.86	85.77	72.97
275	55.77	63.14	51.24	0.01142	49.11	59.46	41.83
276	75.95	80.21	73.19	0.0129	77.81	85.54	71.35
277	56.79	63.88	52.41	0.01145	50.80	60.70	43.68
278	76.90	81.10	74.12	0.01241	78.12	85.79	71.70
279	54.48	62.21	49.79	0.01128	47.18	58.19	39.68
280	76.42	80.73	73.58	0.01277	77.55	85.34	71.06

Table 5.2. Continued

Epoch	Training (Pure) - Initial Score				Test - Initial Score		
	F-Score	Precision	Recall	Cost Function	F-Score	Precision	Recall
281	51.88	60.75	46.71	0.01031	43.51	56.75	35.27
282	76.68	79.80	74.61	0.01209	78.78	85.03	73.39
283	56.36	63.94	51.88	0.01212	49.24	60.28	41.62
284	76.64	80.98	73.80	0.01212	78.18	85.85	71.77
285	54.83	63.09	49.86	0.01149	46.75	58.19	39.07
286	76.42	80.36	73.78	0.01191	77.69	85.19	71.40
287	56.92	64.32	52.44	0.01172	50.49	60.37	43.39
288	76.86	81.14	74.01	0.01179	77.52	85.41	70.96
289	55.33	63.63	50.37	0.01103	46.46	58.64	38.47
290	76.83	80.53	74.35	0.0124	77.67	85.11	71.43
291	58.63	65.45	54.41	0.01222	52.61	62.24	45.56
292	76.65	81.02	73.71	0.01268	77.19	84.88	70.77
293	53.80	62.39	48.61	0.01092	46.84	59.85	38.47
294	76.51	79.70	74.39	0.01179	78.99	85.48	73.42
295	56.26	63.30	51.92	0.0115	50.77	61.21	43.38
296	76.49	80.75	73.69	0.01268	78.13	85.94	71.62
297	56.41	63.55	51.99	0.0112	49.31	59.81	41.95
298	77.40	81.35	74.78	0.01226	78.30	85.58	72.15
299	53.30	60.93	48.65	0.0111	46.20	57.51	38.60
300	76.76	80.57	74.20	0.01228	78.01	85.68	71.59

Fourth, the training used too high a learning rate value, making the training unstable during the process, and diverged. In the present text detection study case, the Adam optimizer sometimes could not effectively control the learning rate during the training if the learning rate was too large. Therefore, using additional momentum could effectively make the training more stable. However, stable training process did not guarantee that the predictive model obtains high performance from that training. So, the training hyperparameter should be handled with care for each machine learning study case. The F-Score, Prediction, and Recall is illustrated in Figure below. The learning set of 1 to 3 used changing learning rate of each epoch which is related to the momentum application. In the other hand, the learning set of 4 – 6 used constant learning rate, which is 1 E - 3, 1 E-4 and 1E-5 respectively. It is shown that using large value of learning rate made the training was highly unstable.

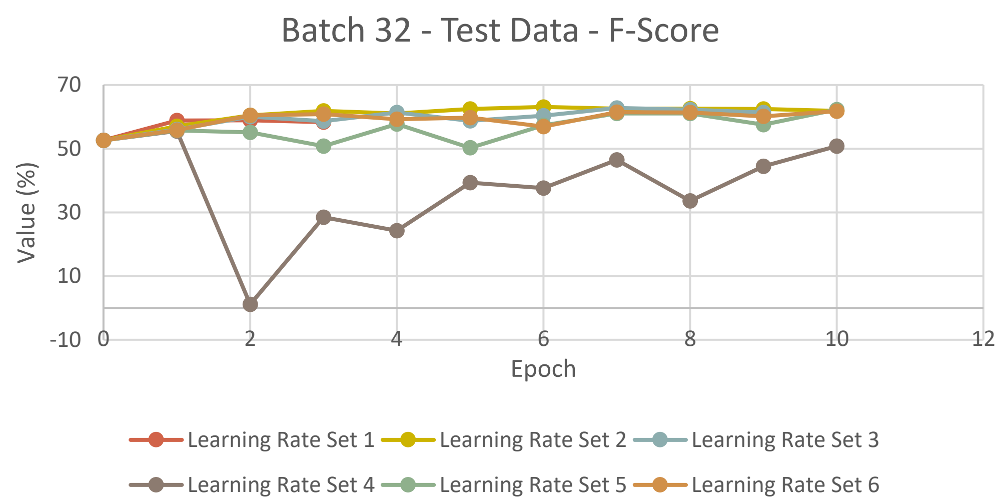


Figure 5.12. F-Score of Batch 32 Training Result – Test Data

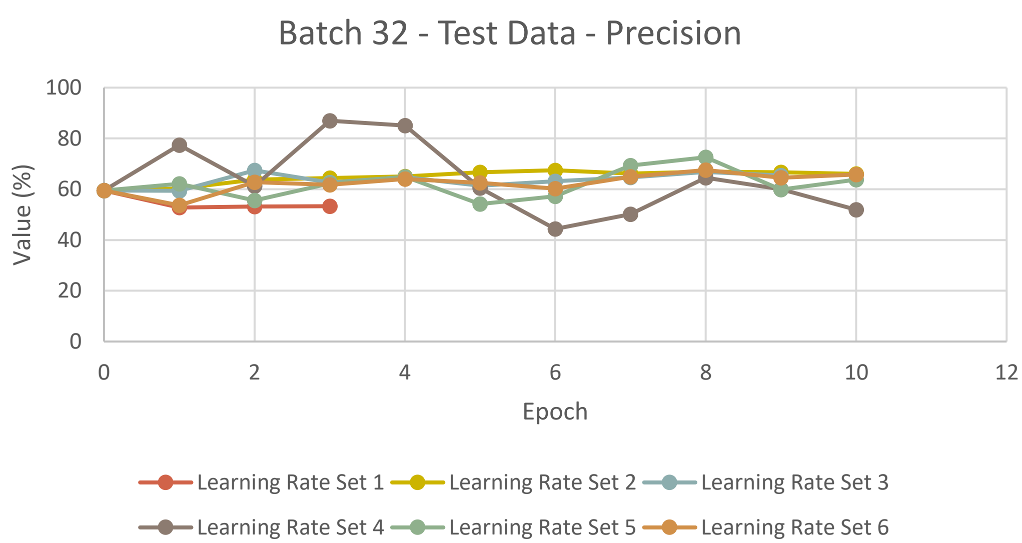


Figure 5.13. Precision of Batch 32 Training Result – Test Data

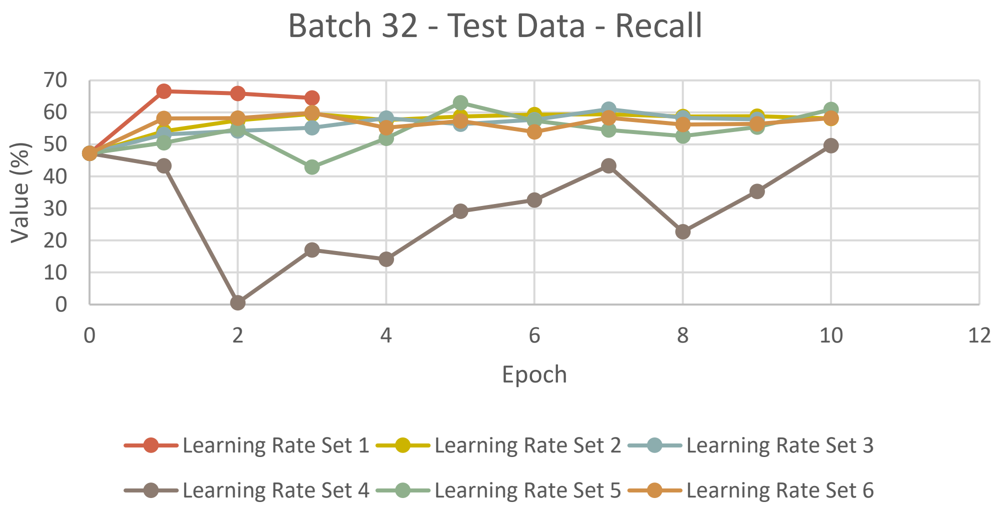


Figure 5.14. Recall of Batch 32 Training Result – Test Data

Lastly, using smaller iterations compared to the number of training images in an epoch was quite useful for searching the good performance of predictive modeling. The trade-off of reducing the number of the iterations was increasing the number of epochs. However, since the training dataset was shuffled for each iteration, it was possible for the model to converge only at most of the test dataset. So, further confirmation of the model by adding more datasets is recommended and unfortunately we did not have the minimum required number of the images to approach the global ground truth.

6. CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

The fine-tuning of the pretrained model to detect the unstructured text in structural drawings were conducted and yielded acceptable performance. Based on the results and lesson learned along the training, some conclusions about the implementation of text detection for reading structural drawings are listed as follows:

- The present predictive modeling workflow and its computational requirement is sufficient for performing unstructured text detection in structural drawings. The present strong-supervised predictive model can be fine-tuned to detect the words that appear in a structural drawing.
- Data augmentations, especially by rotating the words, are important to prepare the model to properly detect the rotated texts. This frequently exists in a structural drawing.
- Partitioning the structural drawing into smaller images is significant fo text detection. Resizing large dimension images into smaller dimensions is necessary to fit the model's input size but can downgrade the input quality.
- Reducing the number of iterations by increasing the number of epochs is effective as a strategy for the predictive model training.
- Training hyperparameters are usually selected and tested with smaller number of training images before executing the full training. This step is helpful to confirm the network architectures and the programming codes are all working well.
- Higher batch size makes the training progress more stable, but it does not influence the performance of the model.

6.2 Recommendations

Some recommendations for further research to expand structural drawing information extraction and improve the current technique applied in text detection are listed as follows:

- Applying text detection and text recognition in hand-written texts would be beneficial since there are a large number of old drawings that need to be digitalized.

- Applying natural language programming to understand the context of the extracted words, so an information summary of the structural drawings can be given to engineers
- Correlating the context of the recognized words and their positions to the nearest objects, such as column labels, wall labels, beam labels, floor plan dimension, drawing scale, structural component dimensions and details, would be beneficial in reading and understanding the structural drawings automatically by the computer.
- Improving the predictive modelling by doing weakly supervised learning to ensure the text detection and text recognition capability. Strong supervisied learning requires character-level annotations at the words, which needs to be partially automated and confirmed by the labeler because the number of characters inside a structural drawing is large.
- Combining the detected text bounding boxes in cropped images into the original structural drawings so the text detection can be processed further into another post-processing.

APPENDIX A. LICENSES AND PERMISSIONS

CRAFT Main Program by Naver Corp.

Source: https://github.com/clovaai/CRAFT-pytorch/blob/master/LICENSE

Copyright (c) 2019-present NAVER Corp.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CRAFT Implementation by Mayank Kumar Singh.
Source: https://github.com/autonise/CRAFT-Remade/blob/master/LICENSE

SynthText Dataset by Ankush Gupta et al.

Source : https://www.robots.ox.ac.uk/~vgg/data/scenetext/

You (the "Researcher"), have requested permission to use the SynthText in the Wild database (the "Database") at the University of Oxford. In exchange for such permission, the Researcher hereby agrees to the following terms and conditions:

1. Researcher shall use the Database only for non-commercial* research and educational purposes.
2. University of Oxford makes no representations or warranties regarding the Database, including but not limited to warranties of non-infringement or fitness for a particular purpose.
3. Researcher accepts full responsibility for his or her use of the Database and shall defend and indemnify University of Oxford, including their employees, Trustees, officers and agents, against any and all claims arising from Researcher's use of the Database, including but not limited to Researcher's use of any copies of copyrighted images that he or she may create from the Database.
4. Researcher may provide research associates and colleagues with access to the Database provided that they first agree to be bound by these terms and conditions.
5. University of Oxford reservers the right to terminate Researcher's access to the Database at any time.
6. If Researcher is employed by a for-profit, commercial entity*, Researcher's employer shall also be bound by these terms and conditions, and Researcher hereby represents that he or she is fully authorized to enter into this agreement on behalf of such employer.

Source : https://github.com/ankush-me/SynthText/blob/master/LICENSE

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

REFERENCES

Baek, Y., Lee, B., Han, D., Yun, S., Lee, H. (2019, 3 April). Character Region Awareness for Text Detection. Retrieved From https://arxiv.org/pdf/1904.01941.pdf

Beucher, S. (1994, September). Watershed, Hierarchical Segmentation and Waterfall Algorithm. Retrieved From https://people.cmm.minesparis.psl.eu/users/beucher/publi/ismm94_beucher.pdf

Biswal, A., (2022, May 18). Recurrent Neural Network (RNN) Tutorial: Types, Examples, LSTM and More. Retrieved From https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn

Bontrhon, L., Beck, C., Lund, A., Zhang, X., Cao, Y., Dyke., S.J., Ramirez, J., Mavroeidis, G.P., Baah, P., Hunter., J. (2021, August 11). Database Enabled Rapid Seismic Vulnerability Assessment of Bridges. Retrieved From https://journals.sagepub.com/doi/10.1177/03611981211032223

Chen, L., Papandreou, G., Kokkinos, L., Murphy, K., Yuille, A. (2017, May 12). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. Retrieved From https://arxiv.org/pdf/1606.00915v2.pdf

Deng, D., Liu, H., Li, X., Cai, D., (2018, 4 January). PixelLink: Detecting Scene Text via Instance Segmentation. Retrieved From https://arxiv.org/pdf/1801.01315.pdf

Deng, J., Dong, W., Socher, R., Li, L., Li, K., Li, F., (2009). ImageNet: A large scale hierarchical image database. Retrieved From https://ieeexplore.ieee.org/document/5206848

Epshtein, B., Ofek, E., Wexler, Y., (2010, 5 August). Detecting text in natural scenes with stroke width transform. Retrieved From https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5540041

Ferreira, J. J., Monteiro, M., (2021). The Human-AI Relationship in Decision-Making: AI Explanation to Support People on Justifying Their Decisions. Retrieved From https://arxiv.org/abs/2102.05460

Gupta, A., Vedaldi, A., Zisserman, A. (2016, 22 April). Synthetic Data for Text Localisation in Natural Images. Retrieved From https://arxiv.org/pdf/1604.06646.pdf

He, T., Huang, W., Qiao, Y., Yao, Jian. (2016, 31 March). Accurate Text Localization in Natural Image Cascaded Convolutional Text Network. Retrieved From https://arxiv.org/pdf/1603.09423.pdf

He, T., Tian, Z., Huang, W., Shen, C., Qiao, Y., Sun, C. (2018, 23 March). An end-to-end TextSpotter with Explicit Alignment and Attention. Retrieved From https://arxiv.org/pdf/1803.03474.pdf

He, W., Zhang, X., Yin, F. Liu, C. (2017, 24 March). Deep Direct Regression for Multi-Oriented Scene Text Detection. Retrieved From https://arxiv.org/pdf/1703.08289.pdf

He, P., Huang, W., He, T., Zhu, Q., Qiao, Y., Li, X. (2017, 1 September). Single Shot Text Detector with Regional Attention. Retrieved From https://arxiv.org/pdf/1709.00138.pdf

Huang, T. S. (2009, 20 April). Computer Vision: Evolution and Promise. Retrieved From http://cds.cern.ch/record/400313/files/p21.pdf

Hu, H., Zhang, C., Luo, Y., Wang, Y., Han, J., Ding, E., (2017, 22 August). WordSup: Exploiting Word Annotations for Character based Text Detection. Retrieved From https://arxiv.org/pdf/1708.06720.pdf

Ioffe, S., Szegedy, C. (2015, 2 March). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Retrieved From https://arxiv.org/pdf/1502.03167.pdf

Jiang, Y., Zhu, X., Wang, X., Yang, S., Li, W., Wnag, H., Fu, P., Luo, Z. (2017, 30 June). R2CNN: Rotational Region CNN for Orientation Robust Scene Text Detection. Retrieved From https://arxiv.org/ftp/arxiv/papers/1706/1706.09579.pdf

Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., Bigorda, L., Mestre, S., Mas, J., Mota, D., Almazan, J., Heras, L. (2013, 15 October). ICDAR 2013 Robust Reading Competition. Retrieved From https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6628859

Karatzas, D., Gomes-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V., Lu, S., Shafait, F., Uchida, S., Valveny, E. (2015, 23 November). ICDAR 2015 Competition on Robust Reading. Retrieved From https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7333942

Kaur, T., Ghandi, T. (2020, 12 March). Automated brain image classification based on VGG-16 and transfer learning. Retrieved From https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9031952

Kingma, D., Ba, J. (2017, 30 January). Adam: A Method For Stochastic Optimization. Retrieved From https://arxiv.org/pdf/1412.6980.pdf

Lavrentyeva, Y., (2021, May 26). Top Computer Vision Applications in Different Industries Right Now. Retrieved From https://itrexgroup.com/blog/computer-vision-applications-in-different-industries/

LeCun, Y., Bengio, Y., Hinton, H. (2015, 28 May). Deep Learning. Retrieved From https://www.cs.toronto.edu/~hinton/absps/NatureDeepReview.pdf

Li, F., Johnson, J., Yeung., (2017, May 10). Detection and Segmentation. Retrieved From http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf

Liao, M., Zhu, Z., Shi, B., Xia, G., Bai, X. (2018, 14 March). Rotation-Sensitive Regression for Oriented Scene Text Detection. Retrieved From https://arxiv.org/pdf/1803.05265.pdf

Liao, M., Shi, B., Bai, X. (2018, 27 April). TextBoxes++: A Single-Shot Oriented Scene Text Detector. Retrieved From https://arxiv.org/pdf/1801.02765.pdf

Liao, M., Shi, B., Bai, X., Wang, X., Liu, W. (2016, 21 November). TextBoxes: A Fast Text Detector with a Single Deep Neural Network. Retrieved From https://arxiv.org/pdf/1611.06779.pdf

Liu, Y., Jin, L. (2017, 4 March). Deep Matching Prior Network: Toward Tighter Multi-oriented Text Detection. Retrieved From https://arxiv.org/pdf/1703.01425.pdf

Liu, Y., Jin, L., Zhang, S., Zhang, S. (2017, 6 December). Detecting Curve Text in the Wild: New Dataset and New Solution. Retrieved From https://arxiv.org/pdf/1712.02170.pdf

Liu, X., Liang, D., Yan, S., Dagui, C., Qiao, Y., Junjie, Y. (2018, 15 January). FOTS: Fast Oriented Text Spotting with a Unified Network. Retrieved From https://arxiv.org/pdf/1801.01671.pdf

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., Berg, A. (2016, 29 December). SSD: Single Shot MultiBox Detector. Retrieved From https://arxiv.org/pdf/1512.02325.pdf

Long, J., Shelhamer, E., Darrell, T. (2015, 8 March). Fully Convolutional Networks for Semantic Segmentation. Retrieved From https://arxiv.org/pdf/1411.4038.pdf

Long, S., Ruan, J., Zhang, W., He, X., Wu, W., Yao, C. (2020, 18 August). TextSnake: A Flexible Representation for Detecting Text of Arbitrary Shapes. Retrieved From https://arxiv.org/pdf/1807.01544.pdf

Long, S., He, X., Yao, C. (2020, 9 August). Scene Text Detection and Recognition: The Deep Learning Era. Retrieved From https://arxiv.org/pdf/1811.04256.pdf

Lyu, P., Liao, M., Yao, C., Wu, W., Bai, X. (2018, 1 August). Mask TextSpotter: An End-to-End Trainable Neural Network for Spotting Text with Arbitrary Shapes. Retrieved From https://arxiv.org/pdf/1908.08207.pdf

Lyu, P., Yao, C., Wu, W., Yan, S., Bai, X. (2018, 27 February). Multi-Oriented Scene Text Detection via Corner Localization and Region Segmentation. Retrieved From https://arxiv.org/pdf/1802.08948.pdf

Memon, J., Sami, M., Khan, R., Uddin, M. (2020, 14 August). Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR). Retrieved From https://ieeexplore.ieee.org/document/9151144

Nayef, N., Patel, Y., Busta, M., Chowdhurry, P., Karatzas, D., Khlif, W., Matas, J., Pal, U., Burie, J., Liu, C., Ogier, J. (2019, 1 July). ICDAR2019 Robust Reading Challenge on Multi-lingual Scene Text Detection and Recognition – RRC-MLT-2019. Retrieved From https://arxiv.org/pdf/1907.00945.pdf

Neu, D., Lahann, J., Fettke, P., (2021, 26 January). A systematic literature review on state-of-the-art deep learning methods for process prediction. Retrieved From https://arxiv.org/pdf/2101.09320.pdf

Ronneberger, O., Fischer, P., Brox, T. (2015, 18 May). U-Net: Convolutional Networks for Biomedical Image Segmentation. Retrieved From https://arxiv.org/pdf/1505.04597.pdf

Ren, S., He, K., Girshick, R., Sun, J. (2016, 6 January). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Retrieved From https://arxiv.org/pdf/1506.01497.pdf

Saha, S. (2018, 15 December). A Comprehensive Guide to Convolutional Neural Network – the ELI5 way. Retrieved From https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Sandberg, I. W., Lo, J. T., Fancourt, C. L., Principe, J. C., Katagiri, S., Haykin, S. (2001). Nonlinear Dynamical Systems: Feedforward Neural Network Perspectives.

Shrivastava, A., Gupta, A., Girshick, R. (2016, 12 April). Training Region-based Object Detectors with Online Hard Example Mining. Retrieved From https://arxiv.org/pdf/1604.03540.pdf

Shi, B., Bai, X., Belongie, S. (2017, 13 April). Detecting Oriented Text in Natural Images by Linking Segments. Retrieved From https://arxiv.org/pdf/1703.06520.pdf

Singh, A., Pang, G., Toh, M., Huang, J., Galuba, W., Hassner, T. (2021, 12 May). TextOCR: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. Retrieved From https://arxiv.org/pdf/2105.05486.pdf

Singh, M. K., (2021, 7 December). Implementation of CRAFT Text Detection. Retrieved From https://github.com/autonise/CRAFT-Remade

Simonyan, K., Zisserman, A., (2015, 10 April). Very Deep Convolutional Networks For Large-Scale Image Recognition. Retrieved From https://arxiv.org/pdf/1409.1556.pdf

Smith, Ray (2007, 05 November). An Overview of the Tesseract OCR Engine. Retrieved From https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4376991

Spencer Jr., B.F., Hoskere, V., Narazaki, Y., (2019). Advances in Computer Vision-Based Civil Infrastructure Inspection and Monitoring. Retrieved From https://www.sciencedirect.com/science/article/pii/S2095809918308130

Thakur, R., (2019, 16 August). Step by Step VGG16 Implementation in Keras for Beginners. Retrieved From https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c

Vincent, L., Soilie, P., (1991, June). Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. Retrieved From https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=87344

van Rijsbergen, C. J., (1979) Information Retrieval (2nd Edition).

Wei, J., Zou, K. (2019, 31 January). EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. Retrieved From https://arxiv.org/pdf/1901.11196.pdf

Wei, T., Sheikh, U., Rahman, A. (2018, 31 May). Improved Optical Character Recognition with Deep Neural Network. Retrieved From https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8368720

Yao, C., Bai, X., Sang, N., Zhou, X., Zhou, S, Cao, Z. (2016, 5 July). Scene Text Detection via Holistic, Multi-Channel Prediction. Retrieved From https://arxiv.org/pdf/1606.09002.pdf

Yao, C., Bai, X., Liu, W., Ma, Y., Tu, Z. (2012, 26 July). Detecting Texts of Arbitrary Orientations in Natural Images. Retrieved From https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6247787

Yeum, C. (2016, December). Computer Vision-Based Structural Assessment Exploiting Large Volumes of Images.

Yeum, C., Dyke, S., (2015, 15 May). Vision-Based Automated Crack Detection for Bridge Inspection.

Zhang, Z., Zhang, C., Shen, W., Yao, C., Liu, W., Bai, X. (2016, 16 April). Multi-Oriented Text Detection with Fully Convolutional Networks. Retrieved From https://arxiv.org/pdf/1604.04018.pdf

Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., Liang, J. (2017, 10 July). EAST: An Efficient and Accurate Scene Text Detector. Retrieved From https://arxiv.org/pdf/1704.03155.pdf