SEGMENTATION-FREE LICENSE PLATE RECOGNITION USING DEEP LEARNING

SOO CHING PAU

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Science (Hons.) Software Engineering

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

April 2017

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : _____

Name : Soo Ching Pau

ID No. : 1305682

Date : 31 March 2017

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"SEGMENTATION-FREE LICENSE PLATE RECOGNITION USING DEEP LEARNING"** was prepared by **SOO CHING PAU** has met the required standard for submission in partial fulfillment of the requirements for the award of Bachelor of Science (Hons.) Software Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : _____

Supervisor : Dr. Tay Yong Haur

Date : 31 March 2017

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Dr. Tay Yong Haur for his invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, I would also like to express my gratitude to ePROTEA-FINEXUS(Group) especially my industry training supervisor, Mr. Joel Choo, the Associate Director of EPROTEA-FINEXUS(Group) for his unwavering support in both resources and monetary.

Special thanks to Cheang Teik Koon and Chong Yong Shean, with whom I shared many discussion and insight. Without them, I may go through much more difficulties than now and the progress of this research would be slow and tedious. My gratitude to them for guiding and sharing me their ideas.

Last but not the least, I would express my thanks to my parent and friends for their support and understanding throughout the entire process of the research. Without them, the project would not be truly complete.

SEGMENTATION-FREE LICENSE PLATE RECOGNITION USING DEEP LEARNING

ABSTRACT

Vehicle License Plate Recognition usually did using the traditional three approaches, namely, classical approach, recognition based segmentation and holistic method. These approaches often relying on character level segmentation and often require extensive handcrafted code in order to work fairly well. In this paper, we focus on recognizing license plate number from the real-world cameras. We propose a unified approach that integrates the segmentation and recognition steps via the use of an end-to-end method that operates directly on the image pixels. We train a model using the convolutional neural network (CNN) and Long-Short Term Memory (LSTM) on a VGGNet19 neural network architecture for sequential feature extract purposes. The main advantages of our approach are that it is segmentation-free and with least amount of preprocessing. Our method performs better than other baseline method and achieves state-of-the-art recognition accuracy.

KEYWORD – vehicle license plate recognition, end-to-end recognition, CNN, RNN, LSTM, segmentation-free recognition

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure 5.1.1-2 Colocation Optimization Process Visualization - Final Stage(Spatial Transformer Network, 20016) 49

LIST OF SYMBOLS / ABBREVIATIONS

VLP	Vehicle License Plate
ALPR	Automated License Plate Recognition
VLPR	Vehicle License Plate Recognition
INSEG	Input Segmentation
OUTSEG	Output Segmentation
OCR	Optical Character Recognition
CNN	Convolutional Neural Networks
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Networks
BRNN	Bi-directional Recurrent Neural Networks
LSTM	Long Short-term Memory
BLSTM	Bi-directional Long Short-term Memory
LRCN	Long-term Recurrent Convolutional Networks
CCA	Connected Component Analysis
GPU	Graphics Processing Unit
LCS	Longest Common Subsequence

CHAPTER 1

INTRODUCTION

The rapid growth and advancement opportunity in Intelligent Transportation System(ITS) has attracted considerably research interest in the field of Automatic License Plate Recognition(ALPR). It's potential application, such as in security and traffic control and monitoring has seen an increasing adoption over the past years(Roberts and Casanova, 2012). Other applications for ALPR technology includes parking lot management, automated ticket issuing and toll payment collection.

1.1 ALPR Difficulties

Most of the existing algorithm only work wells under controlled conditions (Hui & Shen 2016). For instance, some systems require sophisticated hardware to capture high-quality images, and others demand vehicles to slowly pass the camera or even at a full stop.

Recognizing license plate in an open environment is challenging, images captured from the surveillance video often face image degradation problem. For instance, varying lighting condition, occlusion, blurring and shadow. Difficulties level increase because the font and layout of license plate were not standardised. A different set of datasets required a different level of fine tuning and image pre-processing in order to achieve state-of-the-art recognition accuracy. The code is said to be unadaptive despite its complexity.

Figure 1.3.1-1VLP Image Degradation Problem and Vary Sizes

Previous work on image recognition commonly extends CNN to transform the license plate into a single-label classification problem by allowing the CNN to classify one character at a time. This gives rise to problems, such as the segmentation point which determine the cutting edge and width of each segment which acts the role in influencing the accuracy of recognition.

In this paper, we focus on recognizing license plate number from the real-world cameras. We propose a unified approach that integrates the segmentation and recognition steps via the use of an end-to-end method that operates directly on the image pixels.

1.2 Aims and Objectives

The goals and objectives of this research project is

(i) To implement a VLPR system using deep learning approach.

(ii) To develop a VLP Recognizer that able to recognize different styles and image noise on license plate without the hassle of model fine tuning and hand engineer.

(iii) To develop a VLP Recognizer that requires least or no preprocesses and segmentation on dataset.

(iv) To achieve an accuracy of above 90% using end-to-end method.

1.3 Scope

This research will primarily focus on a VLP recognition by using a deep learning approach.

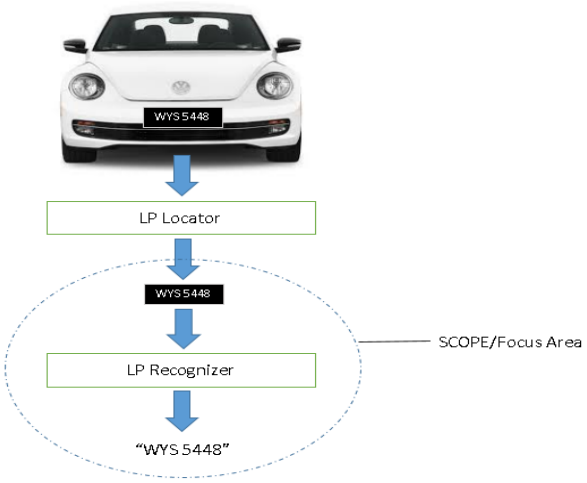


Figure 1.3 -1 Scope

1.3.1 Modules Covered

- VLP Recognition
- Malaysian Private and Commercial VLP numbers
- VLP with maximum of 10 Character in length

1.3.2 Modules Not Covered

- Vehicle detection
- Vehicle License Plate Number detection
- Slanted / Curved Vehicle License Plate number (Motorcycle)
- Specialized Plates
    - Foreign Mission License Plate (39-08-CC)
    - Military License Plate (ZC 8144)
    - Royalties (Tengku Mahkota Johor)
    - Governmental License Plate
    - Trade Plates (W/TP 1544)
    - Trailer Plates (T/BD 6125)
    - Commemorative Plates (PROTON 2020)
    - International License Plate/NON-Malaysian License Plate
    - Illegal License Plate (font size, font style, font faces, character spacing)

CHAPTER 2

RELATED WORK

2.1 Character Segmentation and Recognition

Character segmentation is an operation that seeks to decompose an image consists of a sequence of characters into sub-images of individual symbols. In the paper of Richard and Eric (2006), character segmentation is classified into three approaches,

(a) Classical Approach – Image is cut based on white space or define width.
(b) Recognition based segmentation – System search the image for components that match classes in its alphabet.
(c) Holistic methods – System recognize words as a whole, avoids the needs to segmentation.
(d) Other

2.1.1 Classical Approach

The classical approach can be further broken into the INSEG approach and the OUTSEG approach (Schenkel, et al. 1994).

The INSEG approach is where an algorithms approach is used to determine the possible segmentation point in the sequence of the license plate. Each segment should have equal width and height. Only image point with possible character will be passed into the recognizer and the sequence is generated by concatenation of the character. Algorithms such as Viterbi, are used in determining the output sequence. Some of these algorithms that are used in determine the segmentation point for each character are Connected Component Analysis(CCA) and Projection Analysis.

Theoretically, the background and the characters of the license plate images should be distinguishable through colour. The projection analysis exploits the pixel and binary of the images to determine the top-bottom boundaries and separate the characters. However, the result can be easily affected by rotated images (J. Guo, 2008) (B. Li, 2013). Unlike the connected component which works on rotated images, but performs badly in segmenting characters that are joined together or broken apart.

(S. Chang, 2004). Zheng et al(2007) combine both the projection analysis and connected component analysis to improve the segmentation accuracy.

In contrast to INSEG method, the OUTSEG approach captures many tentative characters by sweeping a small window over the license plate image. Each content of the window is then identify using OCR. The final output sequence is decided by selecting the characters with the highest count consecutively and are segmented by a nil character.

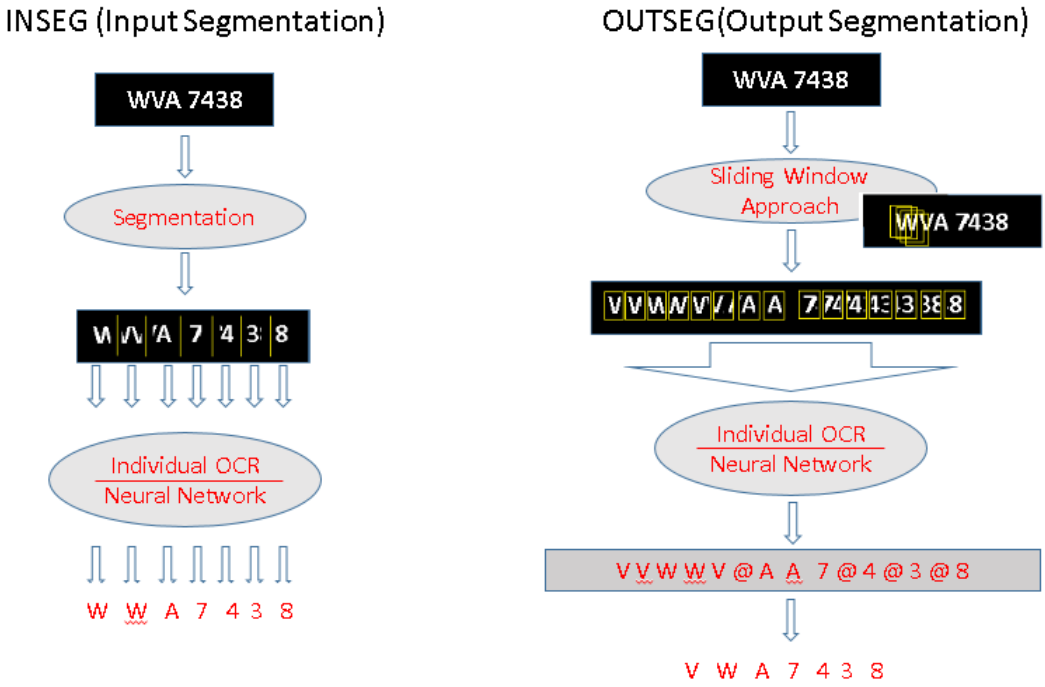
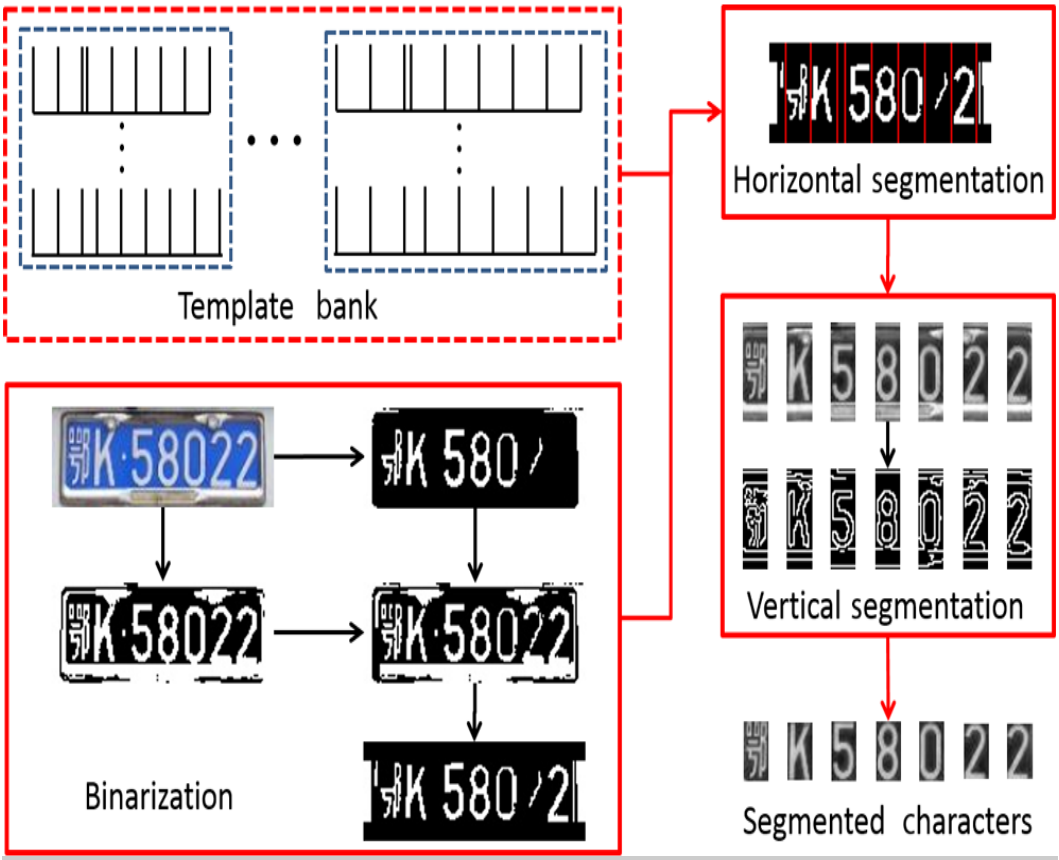


Figure2.1.1-1 INSEG and OUTSEG approach

However, INSEG and OUTSEG approach both faced the limitation where heuristics and hand engineer is needed to determine the segmentation point in different dataset. Besides, for both INSEG and OUTSEG approach, the context of the full image is unidentified in the condition where only a single frame passed into the recognizer as the recognizer unable to justify unsegmented characters based on previous or future frames.

2.1.2 Recognition based Segmentation

There is researcher who treats license plate recognition problem as kind of image classification task. 36 classes (26 alphabets with 10 digits) of images with label were trained and match with the segmented input to gives the sequence. One of the approaches is call template matching. Template Matching based methods recognise each character by comparing the similarity between character to the template. (Rasheed, et al., 2012) (Goel & Dabas, 2013). The similarity can be measure using Mahalanobis distance, Hausdorff distance, Hamming distance and etc. (Du, et al., 2013). The limitation in recognition based segmentation are it only perform well under single font, fixed size characters and no rotation or broken.



2.1.3 Holistic Method - End-to-end Approach

The holistic method, also known as the end-to-end approach allows a sequence-to-sequence labelling where an input sequence is directly mapped on to an output sequence. This approach has the benefit where no segmentation or framing of an image is required, therefore the loss of context problem faced in INSEG and OUTSEG approach can be avoided. This approach significantly reduces the amount of heuristics require as it allows the machine learning to directly model the entire process.

Connectionist Temporal Classification(CTC) is one of the end-to-end solutions. CTC tend to align each label prediction with the corresponded sequence data implicitly. In the work of Hui and Shen (2016) which focusing on degraded VLP, a model with 6 layers of ConvNet model is combine with Bi-directional Recurrent Neural Network(BRNN) for feature extract and interdependencies capturing between features in the sequence. LSTM is applied for sequence labelling and CTC is used for the purposes of sequence decoding. None of the segmentation or normalization is required in their work and achieving state of the art accuracy.

In the work of Liu & Huang (2015), they compared the performance of VLP recognition based on CNN and MLP. The recognition error did in CNN is half(4.134%) of the recognition error did in MLP(7.8%) for character recognition. Similarly, in the work for Malaysian VLPR done by Radzi & Khalil-Hani (2011), a 5 layers CNN model, using second order back-propagation and total of 700 images train dataset is built using Matlab, they achieved 98.79% of accuracy.

2.1.4 Others

In fact, there are other approaches which has also achieved state of the art accuracy. For instance, in the work of Bharat, et al. (2013), they combine both the neural networks and multi-thresholding techniques and achieve an accuracy rate of 98.4%. Multi-thresholding is simplest because it works by converting each pixel in an image to either a black or white pixel. One of the main disadvantages of Multi-thresholding techniques are images with colourful context is difficult to be classified. The converted context is meaningless in this event.

The another approach is morphological Image processing,. In the work of Sneha(2013), neural networks and morphology are used. Morphological techniques probe a small shape of the template and perform dilation or erosion, opening or closing for the purpose of boundary extraction, Convex Hull or Skeletonization.
Similarly, Anuja (2011) proposed the character geometry method for feature extraction and LVQ neural network for license plate character recognition and achieve recognition accuracy of 94.44%. However, all this method requires extensive data pre-processing and heuristics

2.2 Convolutional Neural Network

Convolutional Neural Network(CNN) are a feedforward neural network uses back-propagation algorithm that designed to use the minimal amount of pre-processing in the event of feature extract of an image. This layer reduces the amount of handcraft codes by creating a more adaptive algorithm in identify the license plate.

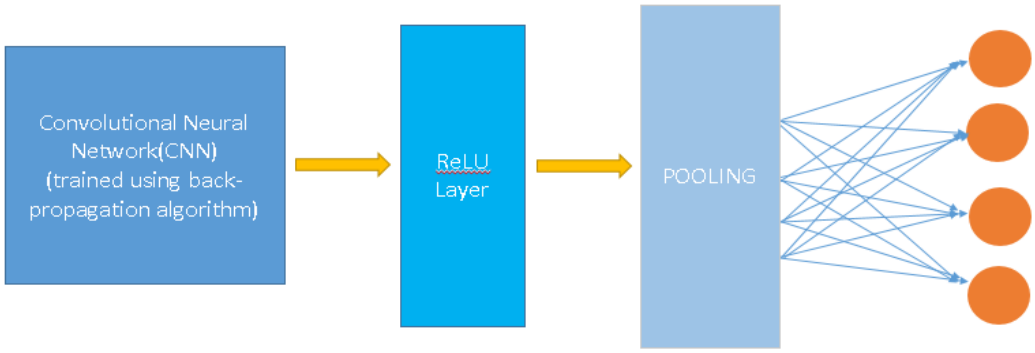


Figure 2.2-1Convolutional Network Architecture

The image will pass as input into the CNN layer for feature extraction, known as forward propagation. Follow by the Backward propagation is to minimize the error. Rectified Linear Unit(ReLU) layer will perform element-wise activation. The activation function is to increase the non-linearity. ReLU transforms the input so that it is separable. (Karpathy, et al. n.d.) The advantages of ReLU layer is that it works well without pre-training and can run very quick and accurate compare to other function. (Liu, et al. 2015) The pooling layer then progressively reduce the dimensionality of the image to control overfitting.

However, to train the CNN with good accuracy and high recognition rate, high amount of training labelled data is needed as a CNN consists of multiple layers of neurons (Hui and Shen 2016) (Liu, et al. 2015).

2.3 Recurrent Neural Network/Long Short-term Memory

Long short-term memory is one of the recurrent neural network architecture. The main features in LSTM is that it excels at remembering values or pattern for long or short duration of time compare to RNN which only able to make use of the previous context. The reason for this is mainly due to its structure. The repeating module for LSTM consist of 4 layers, interacting in a special way within the chain like structure. However, for RNN, there is only single neural network layer within the repeating modules.

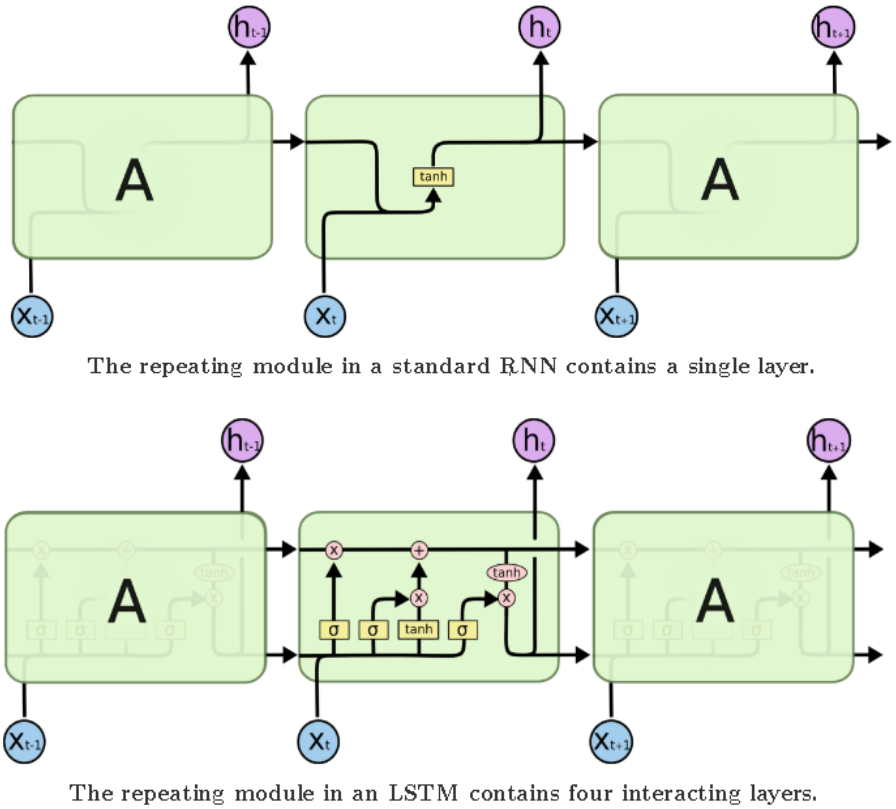


Figure 2-2-1Architecture View of RNN and LSTM

2.4 Facts Finding

Table 1Table of Features by Approaches

Approaches / Features	INSEG Approach	OUSEG Approach	End-to-end Approach
Image Pre-processing	Optional	Optional	Optional/Not Required
Segmentation	Mandatory	Mandatory	Not Require
Manual Sequence Alignment	Mandatory	Mandatory	Not Require
Dataset Fine-tuning	Require	Require	Not Require

Table 2 Pros and Cons of Neural Network Techniques

	Advantages	Disadvantages
CNN	Feature Extract and Learning	Require huge amount of data for training
RNN	Support time-series data and possess internal memory for storing information and data	Only able to store 1 layer of data and information
LSTM	Able to store multiple layers of data or information and better at finding and exploiting long range context	Do not support sequence transcription task, need to align the input to the target sequences

CHAPTER 3

PROPOSED SOLUTION

We present two (2) different combination of neural network approach for VLP recognition problem, a CNN-RNN model and a CNN-LSTM model. Both approaches avoid the need for character segmentation and pre-or post-processing. The CNN-RNN and CNN-LSTM models is trained using 15-layer VGGNet architecture and 19-layer VGGNet architecture for comparison. We keep the VGGNet architecture and other hyperparameters unchanged in our implementation to ensure its simplicity.

3.1 Problem Description

In this experiment, we restrict our problem to only recognise typical Malaysian VLP. Malaysian VLP is generalised into the following pattern:

S(C)(C) N (C)

where

S = Starting character in the list ABCDFHIJKLMNPRSW

C = character range from A-Z, brackets denote optional character.

N = number from 0 to 9999

End to end approach require the output sequence length to be bounded, therefore the VLP labels will be padded with zeros at the front of the sequence in order to have the same length. For instance, VLP ‘MAR5052’ will be padded with three (3) null character to become ∅ ∅ ∅ MAR5052. Most of the license plate number contains less than or equal to eight (8) characters, so we may assume the sequence length n is at most a value of N =10, which is sufficient for our experiment.

3.2 System Architecture

Our VGGNet configuration is quite different from the one used in the top-performing entries of ILSVRC-2013 competition. There is a total of 12 layers of ConvNet and 3 Fully-connected layers in VGGNet15, while 16 layers of ConvNet and 3 Fully-connected layers in VGGNet19. A batch normalization process is added after each Convolutional Layer because it was shown to be able to speed up the training and give a boost to the performance.

We incorporate a single non-linear rectification layers(ReLU) after each and every batch normalization process to perform element-wise activation. ReLU transforms the input so that it is separable. A MaxPooling layer is inserted whenever there is an increase in the depth of ConvNet so that it progressively reduces the dimensionality of the image to control overfitting. All the other hyperparameters of the VGG network remain unchanged in order to keep the whole train model simple and practical.

Table 3 Convolutional Neural Network Configuration

Convolutional Neural Network Configuration	
VGGNet-15	VGGNet-19
15 weight layers	19 weight layers
Input (120x240 RGB image)	
conv3-64 conv3-64 conv3-64	conv3-64 conv3-64
MaxPooling	
conv3-128 conv3-128	conv3-128 conv3-128
MaxPooling	
conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
MaxPooling	
conv3-256 conv3-256	conv3-512 conv3-512 conv3-512 conv3-512
MaxPooling	
conv3-256 conv3-256 conv3-256	conv3-512 conv3-512 conv3-512 conv3-512
MaxPooling	
FC-8192	FC-16384
FC-512	
FC-360	

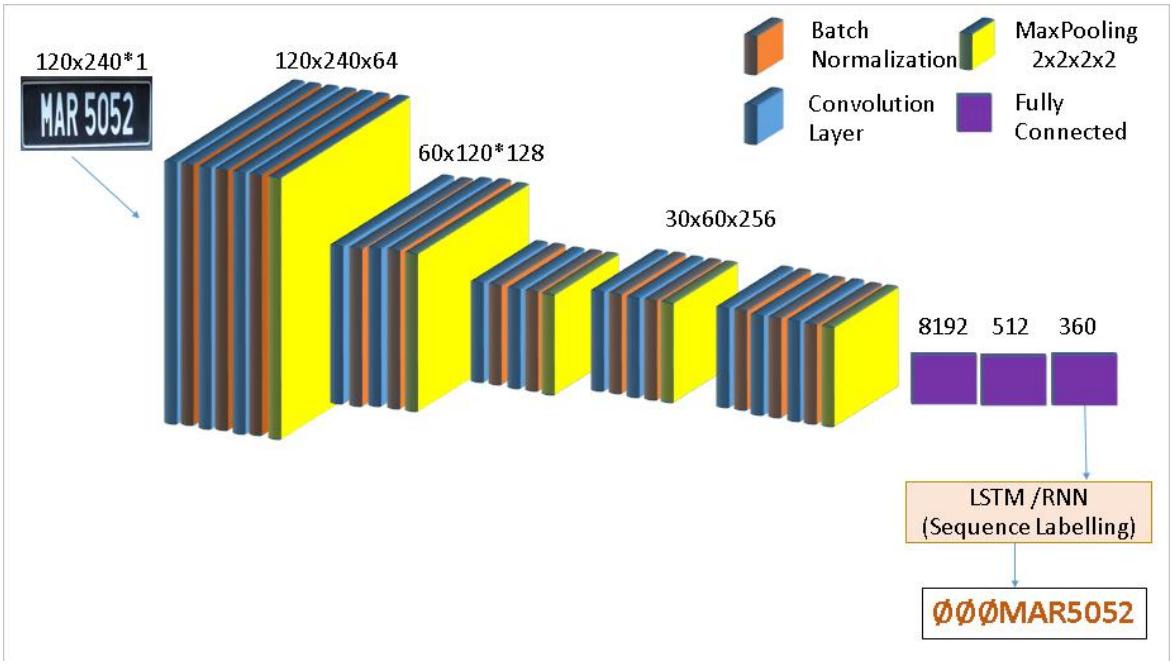


Figure 3.2-1 VGG15 Network Architecture

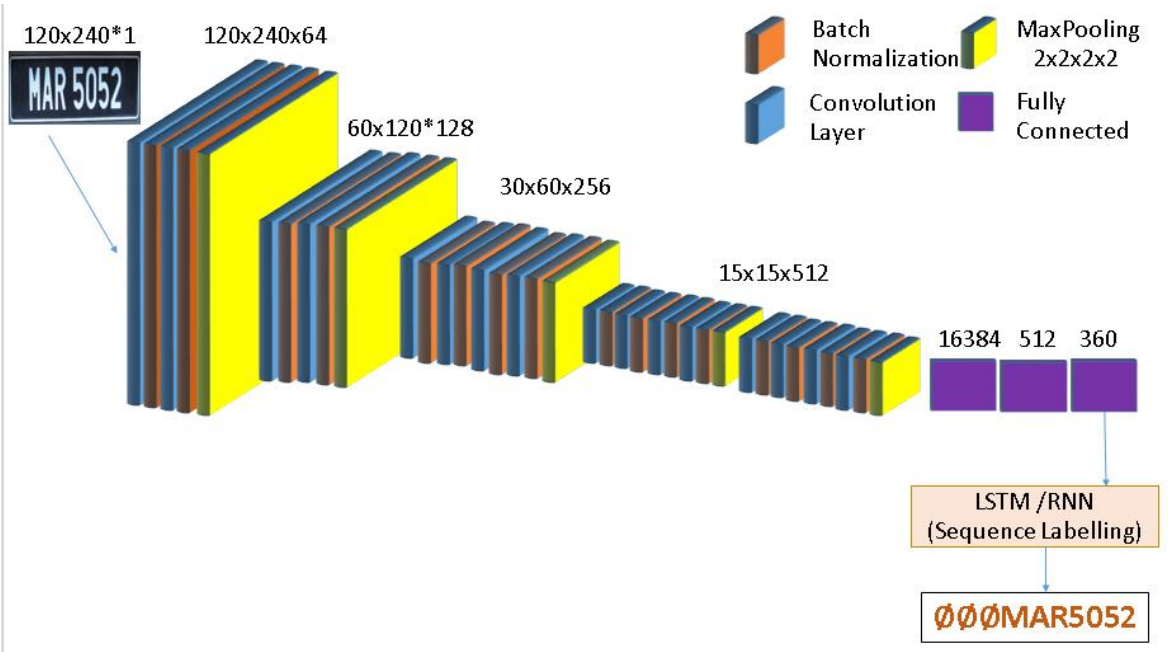


Figure 3.2-2 VGG19 Network Architecture

3.2.1 Features Learning and Sequence Labelling

The images are fixed at a size of 240 widths by 120 height in order to input into the ConvNet in VGGNet. We extract only one out of three of the dimension in RGB value to make the training more efficient.

. Unlike a usual multi-class classification task which output the probability of each class label, we modify and transform this into a sequence labelling cases by allowing the network to directly output the entire sequence in one forward pass of the input image.

An architecture similar to the paper of Goodfellow, et al. is used, where each character in the sequence is represented by X neurons in the output layer, resulting in a total of K × X number of neurons in the output layer. We assign the first X number of output neurons to the first character of the sequence, the second X number of neurons to the second character and so on.

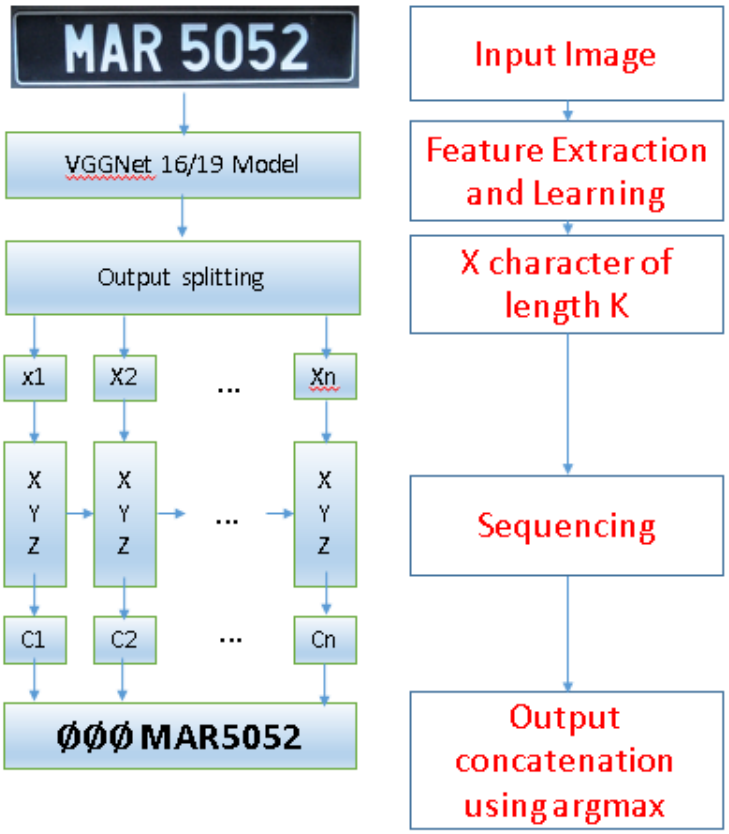


Figure3.2.1-1 Sequence Labelling Based Plate Recognition

Refer to figure 3.2.1, the outputs from the ConvNet are not directly take as the final result. Instead, they are passed into either RNN or LSTM, depending on the experiment, for sequence labelling. The padded sequencing result is final.

3.3 Technology/Techniques Involved and Rationale

3.3.1 Deep Learning Approach and Architecture

Conventional techniques and approaches that focus on character segmentation based plate recognition often require different level of handcrafted engineer and heuristics. Redundant code fine tuning when a new pattern of dataset is added are unavoidable. The combination of CNN to RNN/LSTM approach allow the hidden units of CNN to have access to the features in the entire image and output the entire sequence in one forward pass, instead of only seeing part of the window image and perform probability classification.

The deep learning model is then plugged into VGGNet architecture for training purposes. VGGNet is chosen among the others mainly because of its simplicity and depth. VGGNet achieves incredibly accuracy because the filter size of the VGGNet is smallest among the other architecture, with the size of 3x3 and stride and pad of 1. Thus, the feature maps extracted would be very fine.

3.3.2 Batch Normalization

Normalization techniques are often used for pre-processing to make the dataset comparable across the feature. However, we do not apply normalization for pre-processing in this paper, but attached to the deep neural network. Training deep neural network is complicated because the distribution of each layer's input parameter constantly changing during training across the model. Ioffe and Szegedy refer this as Internal Covariate Shift. Parameter Initialization slows down the training and require lower learning rates in order to learn multi-parameter. Batch Normalization allows us to use higher learning rates and regularize the gradient. Resulting in an acceleration of the learning process. (Ioffe & Szegedy, 2015)

3.3.3 Data Augmentation

Data augmentation is the act of artificially inflating the dataset size by performing various transformations onto the dataset, thereby generating new, distinct data while preserving the label. Data augmentation is performed to tackle low number in dataset. This technique shows the best effect when the network is deep, but dataset number is low, it gives a boost to the performance and minimizes the possibility of overfitting (Wang , 2015).

Usually, Artificial variation is inserted into the original dataset to generate a new set of data. The artificial variation here refers to the random rotation, random translation, scaling, or by adding addictive Gaussian noise, salt pepper noise and Poisson noise. Data augmentation has proven to be effective in improving the accuracy of the trained model as in the case of Choo, et. al. (2012).

3.3.4 Using GPUs

The depth of the volume increases because the number of filters increases as it goes down the model (filter size of 3x3). Besides, the number of filters doubles after each maxpool layer. The reason is because shrinking of dimensions, but growing in depth. The computational power requires will shoot as the depth of the model increase. NVIDIA GPU, particularly CUDA which support parallel computing enables boosting of computing performance by harnessing the power of the GPU. For instance, during the experiment, around 27,000 images were passed into the model, each epoch took around 6hours to train by using CPU, but only took 15 minutes to train by using GPU.

CHAPTER 4

EXPERIMENT

4.1 Evaluation Criteria

The most direct way to evaluate the performance of the model is by computing the labels that are perfectly predicted. However, doing so would not take into account the performance of the model in recognising individual character. Therefore, we introduce the Levenshtein Distance and average ratio metrics analysis in order to evaluate the performance per character.

4.1.1 Levenshtein Distance

Levenshtein Distance is the measure of the similarity between source and target. The distance also calls as the edit distance, refer to the number of deletion, insertion or substitution required to transform the target to the source.

For example, computing the Levenshtein distance of labelled VLP "WWW1111" and predicted VLP "WVV1171" would be 3, this is because substitution of character VV and 7 are required in order to transform the predicted VLP to labelled VLP. The formula for Levenshtein distance is given as follow,

$$\operatorname{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \operatorname{lev}_{a,b}(i-1,j) + 1 \\ \operatorname{lev}_{a,b}(i,j-1) + 1 \\ \operatorname{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

Figure4.1.1-1Levenshtein Distance Formula

Where

a = a sequence of length |a|

b = a sequence of length |b|

$\operatorname{lev}_{a,b}(i,j)$ = the Levenshtein distance between the first i characters of a and the first j characters of b

$1_{(a\_i \neq b\_j)}$ = the indicator function equal to 0 when $a_i = b_i$ and equal to 1 otherwise

4.1.2 Ratio Metric Analysis

Ratio metrics analysis is to calculate the ratio between the source and the target. In ratio metric analysis, the formulae given is as below,

$$\mathbf{((M+N) - \textit{distance}) / (M+N)}$$

where,

M, N= length of source, target,

distance = LCS distance.

The Longest Common Subsequence(LCS) is to calculate the common individual character appear in both the source and target. if the length of predicted output is one longer than the length of source input, the LCS distance is penalized by 1.

Table 4 Comparison of Evaluation Methods

Source Input	Target Output	Perfect	Levenshtein Distance	LCS	Ratio Metrics Analysis
Singing	Singing	1	0	0	1.00
Singing	Sittiang	0	3	6	0.60
Singing	_ _ _ _ _ _ _	0	7	7	0.00
Singing	g	0	6	6	0.25

4.2 Dataset

4.2.1 Train Set

We received about a total of 46,000 unprocessed license plate images at a size of 240px width x 120px height. We performed dataset filtering by removing dataset that are irrelevant to Malaysian vehicle license plate pattern and images that are unable to be recognise at human level. There is a total of about 30,000 images is usable for the purpose of training. In our experiment, we will train the model separately using un-augment trainset and augmented trainset to compare the performance.

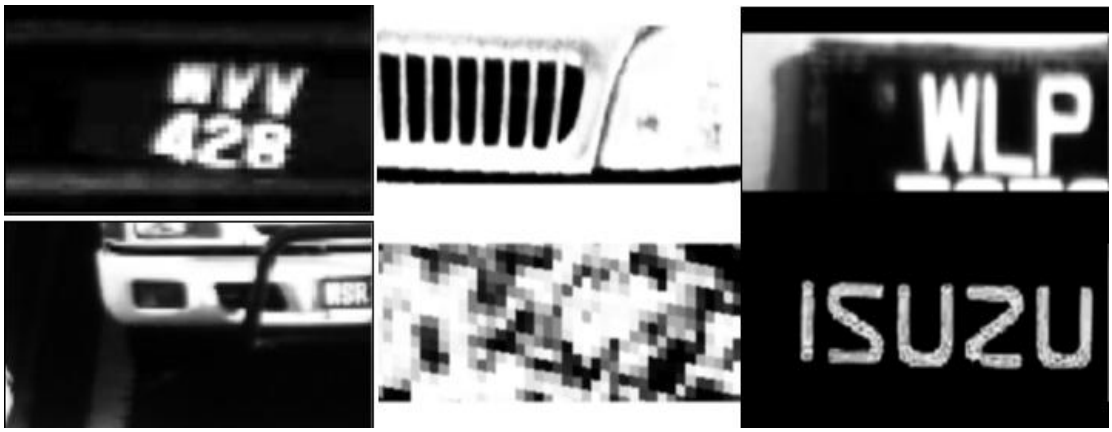


Figure 4.2.1-1Example of Filtered Dataset

Figure 4.2.1-2Example of Train Dataset

We perform data labelling by label all the license plate images with correct sequence and concatenate the license plate number to the end of the image file name. For instance, image filename “20160129_085555.479301_2.jpg” and VLP label “WVA1543” become “20160129_085555.479301_2_WVA1543.jpg”.

We augment the data and the dataset increase to about 310,000 license plate images. The augmentation techniques that used in our train dataset includes,

- Scaling

  Since the dataset contains images of slightly different scale, we augment the data by scaling images randomly by 5% to 30%.

- Translational

  Some of the images from the dataset can be seen to have large positional offsets. Random translations from -1 pixels to 1 pixels on the x-axis and -12 pixels to 12 pixels on the y-axis were performed.

- Rotational

  The dataset images were not aligned properly. The images could be slightly skewed or slanted. Although the nature of CNNs does provide some rotational invariance, we augment the dataset by randomly rotating the images between 2 degrees to 5 degrees.

- Blurring/Sharpening

  Blurring is done by convolving the image with a Gaussian kernel and using the output from the kernels as the smoothed image. Sharpening is done by subtracting an image with a smoothed image.

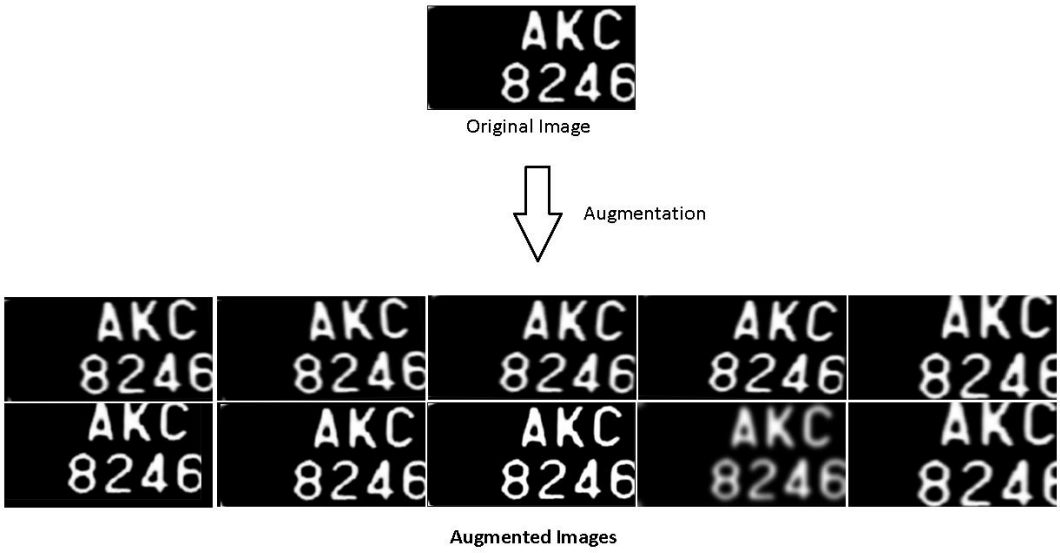


Figure 4.2.1-1 Before Augmentation and After Augmentation

4.2.2 Test Set

To evaluate the performance of the model, we run the benchmarking using 3 folders of image dataset, namely LPR44, LPR45 and an open environment dataset. The image quality level decrease across the dataset and the noise level increase across the dataset. There is a total of 409 images, 737 images and 2800images in the test folder LPR44, LPR45, and open environment dataset respectively.

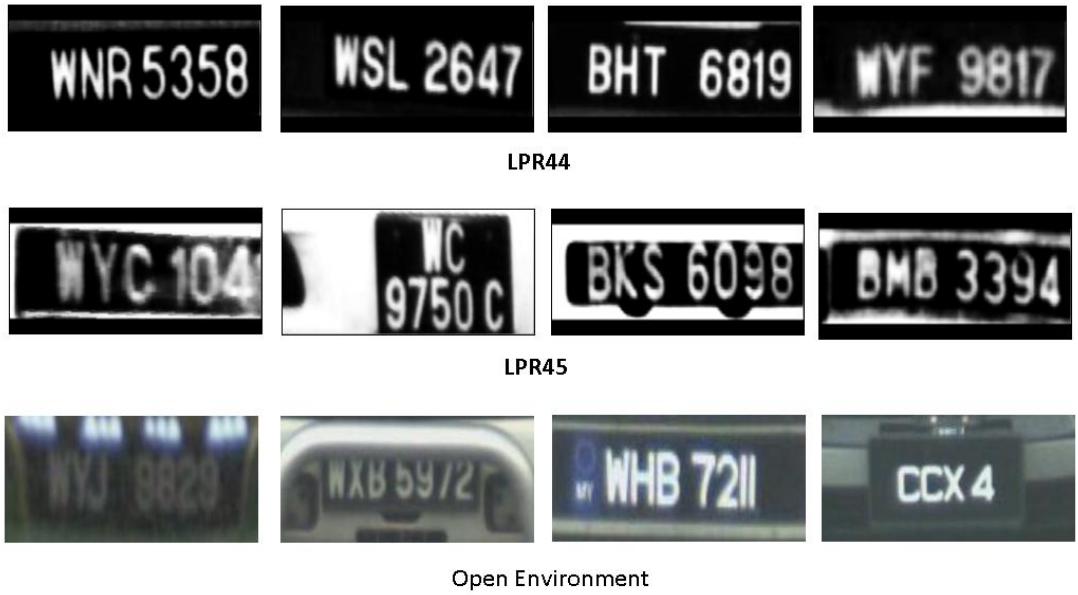


Figure 4.2.2-1 Sample of Test Set

4.3 Result

Data Variation	Neural Network Technique	TEST SET			
		LPR44 [Difficult Level -Easy] - %	LPR45 [Difficult Level -Medium] - %	Open Environment Dataset [Difficult Level -High]	
				Non-Localised - %	Localised - %
Un-augment	VGG15 + RNN	92.42	76.26	8.64	-
	VGG15 + LSTM	0.73	0.81	0.03	-
	VGG19 + RNN	95.84	83.45	15.16	-
	VGG19 + LSTM	7.58	4.61	0.14	-
Augmented	VGG15 + RNN	97.79	93.62	50.10	41.94
	VGG15 + LSTM	97.31	93.49	52.96	42.02
	VGG19 + RNN	97.31	94.43	63.24	52.06
	VGG19 + LSTM	**98.04**	**94.84**	**65.96**	**54.85**
Synthesis Dataset	VGG15 + RNN	29.756	16.96	7.18	-

Table 5Recognition Accuracy of All Experiment done

4.4 Discussion

The performance of all experiments done are tabulated in table 4. The recognition accuracy shows the highest in LPR44, follow by LPR45 and open environment dataset. This is because the image quality decrease across the test set, more noise especially blurring is introduced into the test license plates.

4.4.1 Neural Network Techniques

In our experiment on the model that train with augmented train dataset, LSTM performed slightly better compare to RNN. Averagely, LSTM gives about additional 2percent accuracy boost than RNN. One of the reasons for this is because LSTM is better at handling long-term dependencies. This makes LSTM perform better in connecting previous learned features to the present recognition task compare to RNN. Theoretically, the hyper-parameters of RNN can also be fine-tuning in order to handle long-term dependencies, but as the dataset grow, RNN becomes difficult to learn to connect all the features learned.

However, focusing on the un-augment segment, it is observed that the model train using RNN are outperformed compare to the model that are trained using LSTM, which is diverged from what we claim in the last paragraph. This is because LSTM, the structure itself is a chain of repeating module of RNN, this causes the LSTM require a higher amount of train data to feed the model compare to RNN. Compare to the augmented train dataset, the data volume for training in un-augment train dataset is only about 1/10 of the augmented train dataset. This resulted the LSTM model to capture lesser features compare to RNN model.

4.4.2 Depth of Network Architecture

Next, we also investigate the depth of the network in affecting the feature learning abilities of the model. In our experiment, we increase the depth of the network using 2 different depth networks, namely VGGnet15 and VGGNet19 to observe the increase in recognition performance. Our result shows that VGGNet19 can learn better compare to VGGNet15. The reason for this is because the volume depth of VGGNet19 is 2 times higher than the volume depth of VGGnet15. The increase in volume depth indicates the increase in number of filters in each convolutional layer. As the number of filter increases, the most evident benefits are more details in an image can be captured, features learned would increase as well. Besides, there is

additional 4 layers of convolutional layer in VGGNet19 compare to VGGNet15 which helps better feature learning. From table 4, it can see that under augmented, the recognition performance of VGGNet19 is much better than VGGNet15, especially in open environment dataset.

4.4.3 Data Augmentation

Vehicle license plate dataset is difficult to obtain due to the identifiable structure and privacy concern, by using augmentation techniques which generate higher variation dataset, we successfully showcase the augmentation techniques in boosting the performance of the model. High variation in the dataset will allow the model to learn noisy feature which is good in generalising the model. A generalised model is important from the aspect of practical usage because in real case system, we unable to collect all sort of images into our train dataset, therefore a generalised model will be able to recognised VLP better in open environment condition. From table 4, it can see that model trained using augmented dataset perform significantly better compare to the un-augment dataset especially in recognising open environment. Data augmentation techniques also show to be able to improve the recognition accuracy by 2 to 15 percent in controlled test dataset (LPR44 and LPR45).

4.4.4 Character Level Recognition Accuracy

Table 6 Character Level Recognition Accuracy

Neural Network Technique	TEST SET			
	LPR44	LPR45	Open Environment Dataset	
			Non-Localised	Localised
VGG15 + RNN	99.60	98.81	83.19	79.20
VGG19 + RNN	99.61	98.93	86.98	82.60
VGG15 + LSTM	99.54	98.54	84.51	79.67
VGG19 + LSTM	**99.70**	**98.91**	**87.82**	**84.15**

Table 6 shows the character level recognition accuracy by the test set. We able to achieve character accuracy up to 87% in open environment test set. We made some inference from figure 4.4.4-3 Confusion Matrix of non-localised open environment

test set, there is a large number of digits compare to the characters, and this is the reason most wrong prediction is on characters instead of digits. Besides, digit ‘1’ and character ‘o’ occupy a large proportion of the wrong prediction, digit 1 will predict as ‘7’ while character ‘o’ will be predicted as ‘0’.

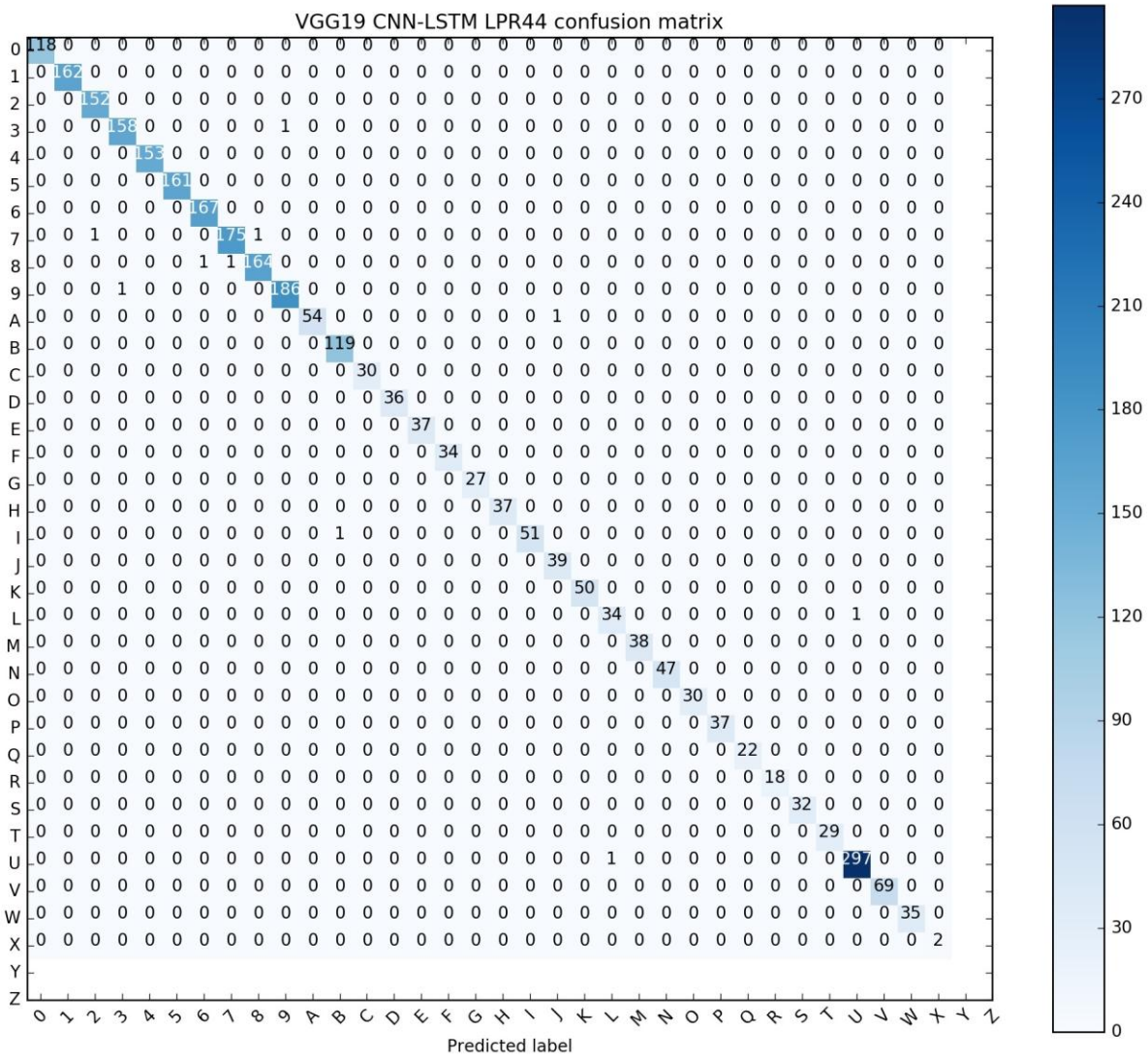


Figure 4.4.4-1VGG19 CNN-LSTM LPR44 Confusion Matrix

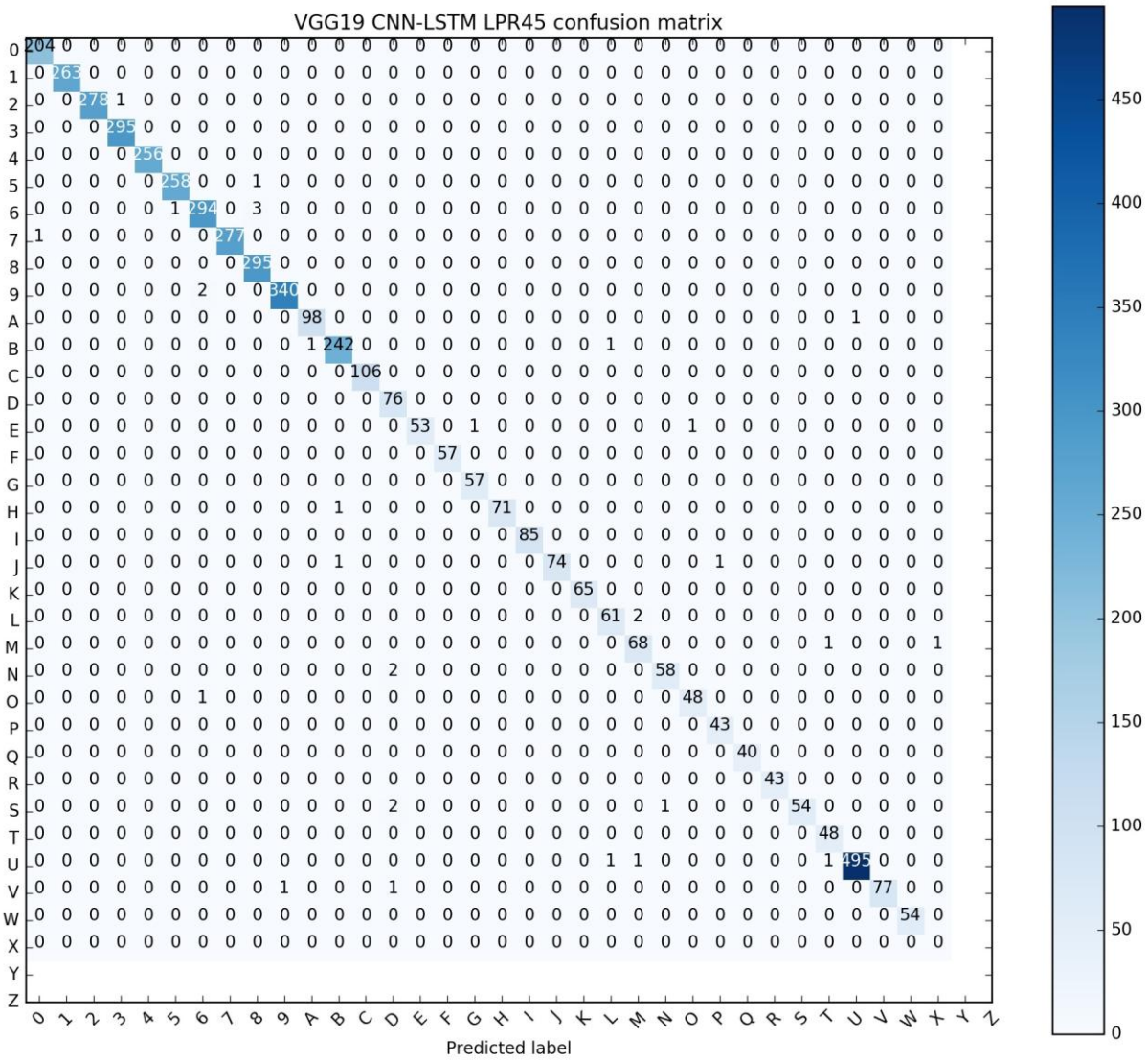


Figure 4.4.4-2VGG19 CNN-LSTM LPR45 Confusion Matrix

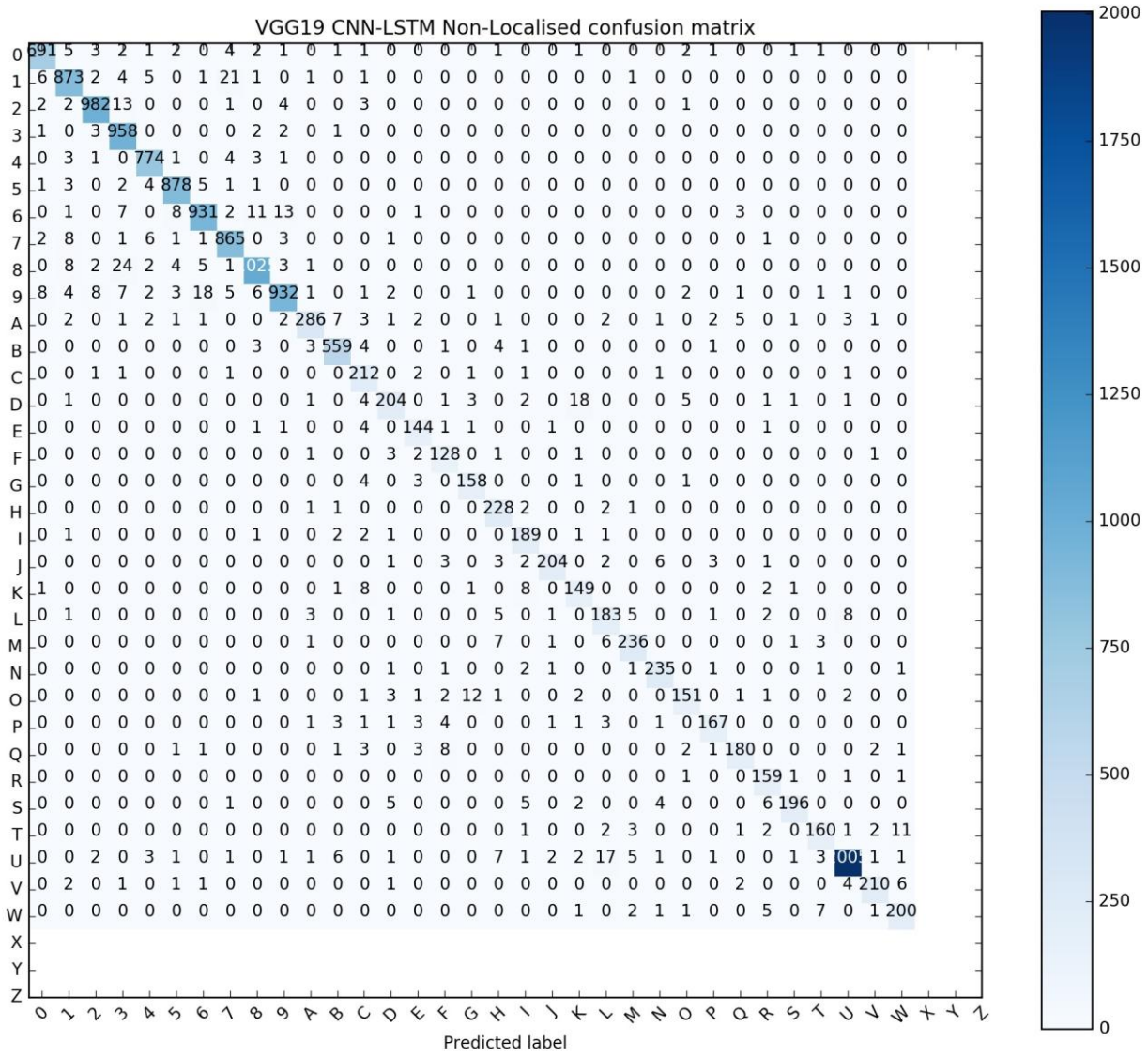


Figure 4.4.4-3VGG19 CNN-LSTM Non-Localised Confusion Matrix

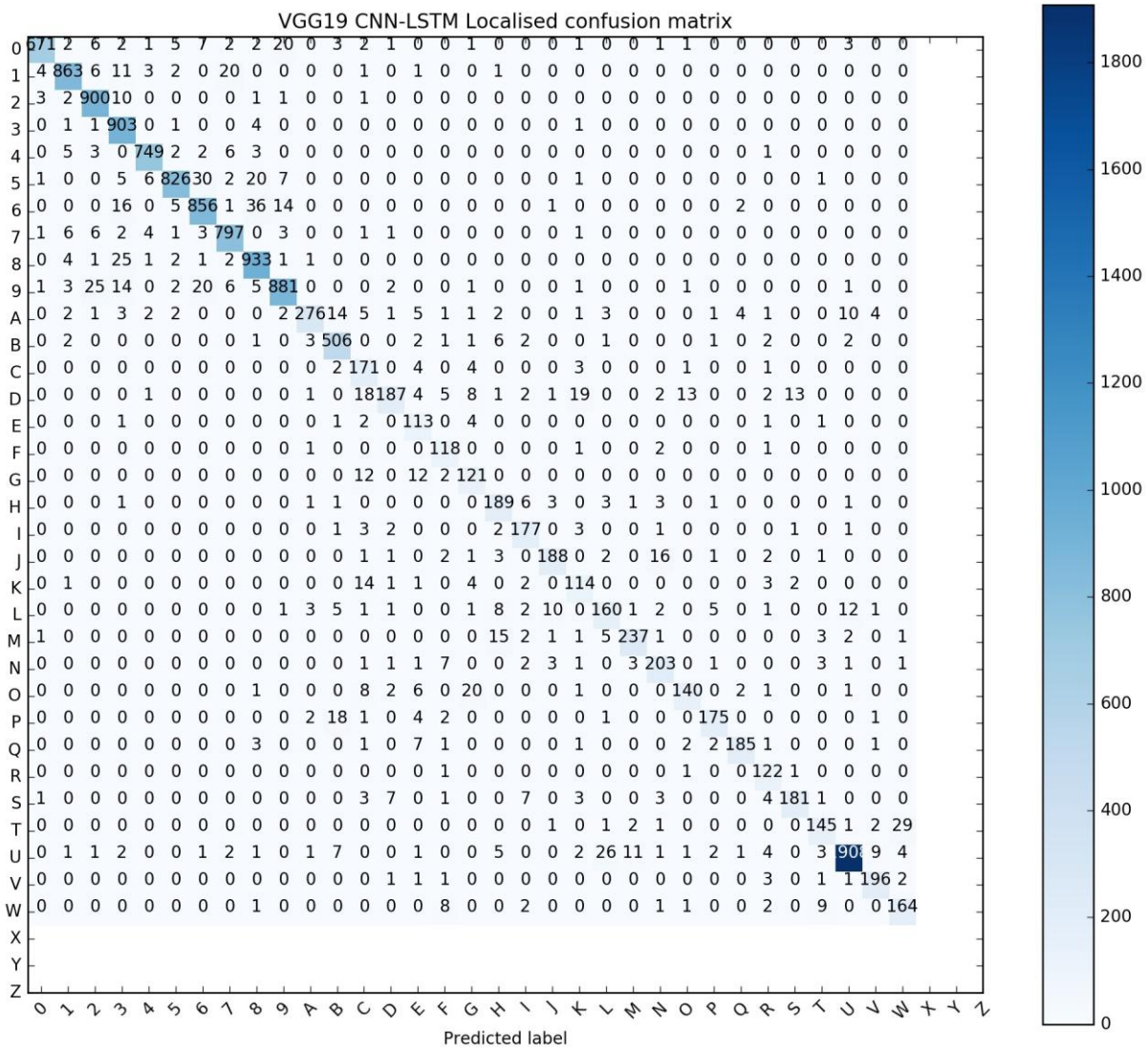


Figure 4.4.4-4VGG19 CNN-LSTM Localised Confusion Matri

4.4.5 Effect on Localised Vehicle License Plate

Our recognition model performs well in the test set LPR44 and LPR45, but not for open environment dataset. We analyse 3 of the test set and figured that there are differences between LPR44, LPR45 to open environment test dataset. VLP in LPR44 and LPR45 are borderless, focused on vehicle license plate number and the background, in either white or black. However, VLP in open environment test dataset come with some background of the vehicles, which act as noise to the image. Refer Figure 4.4.5 for sample images from open-environment test dataset.

In order to boost the recognition performance and determine the root cause, we attempted to compare the recognition performance by localised the test images. We manually crop all the images because it is time-consuming to train a model for cropping purposes. We resize the images using a small python program that create by own.

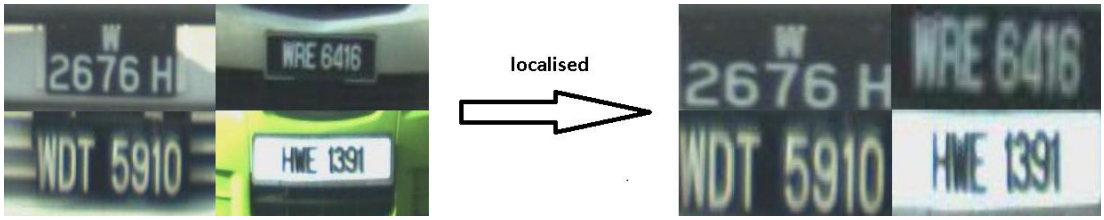


Figure 4.4.5-1Before and After Localise

Table 7 Recognition Performance of Localised and non-localised VLP

Data Variation	Neural Network Technique	TEST SET (Open Environment Dataset)	
		Non-localised	Localised
Augmented	VGG15 + RNN	50.10	41.94
	VGG15 + LSTM	52.96	42.02
	VGG19 + RNN	63.24	52.06
	VGG19 + LSTM	65.96	54.85

Theoretically, removing or minimizing the noise of an image will help improve the model. However, our experiment results on recognise localised open environment dataset shows that remove the background noise of the license plate image do not help in boosting the accuracy of our model. The reasons for this are, the resolution of the image in non-localised dataset is low, thus, when we further cropping and resizing, the bitmap and feature map will change completely.

4.4.6 Synthesis Dataset

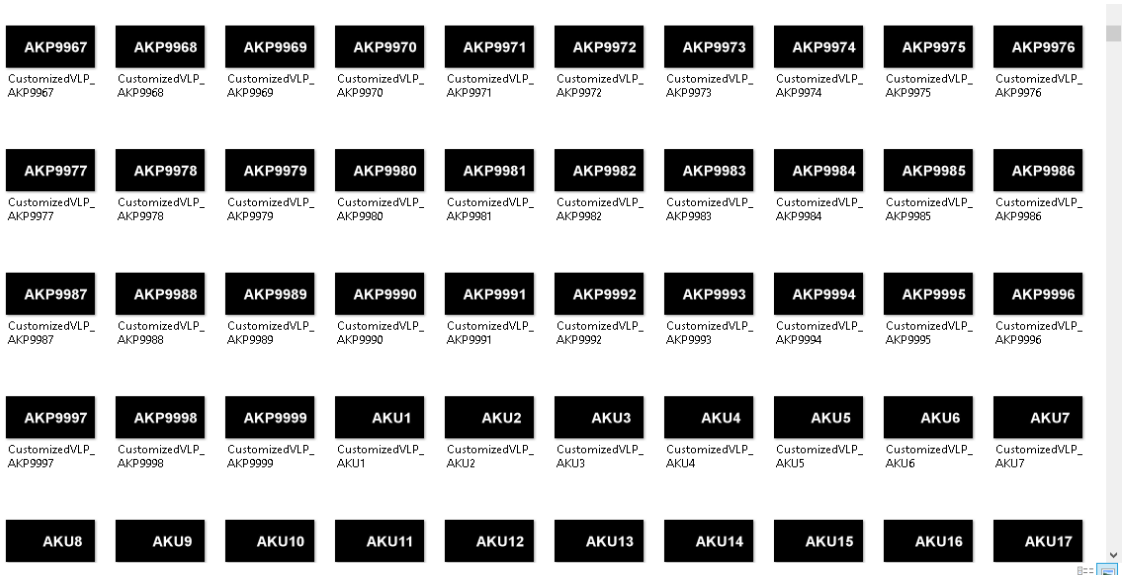


Figure4.4.6-1Sample Synthesis Dataset

The training dataset that uses in our experiment mostly consist of Malaysia City license plate pattern, therefore our model performs badly given the foreign state license plate pattern. In order to have a model with better recognition accuracy, we try to introduce unseen pattern into the model by synthesis dataset by myself using own defined python program. We created about 900,000 images. To ensure the VLP that created by us as similar as the real one, we use the font- Arial as stated in Road Transport Ministry (VLP Rules and Regulation).

Table 8 Performance of Synthesis Dataset

Data Variation	Neural Network Technique	TEST SET		
		LPR44	LPR45	Open Environment Dataset
Synthesis Dataset	VGG15 + RNN	29.756	16.96	7.18

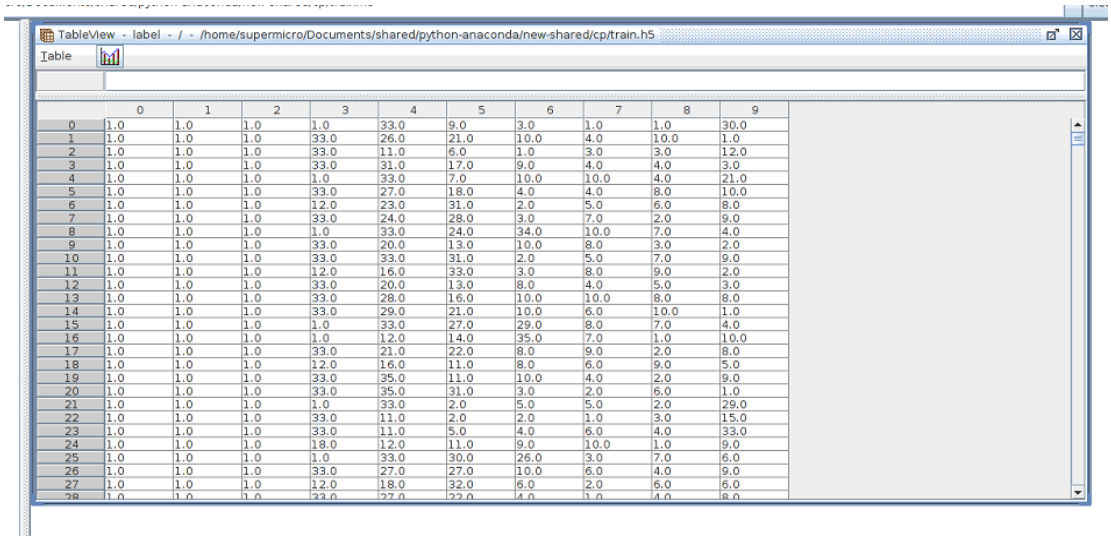
We train the 900,000 images on top of a VGGNet15 RNN model. However, it performs badly. There is few reason why this model performs badly,

- The ‘Arial’ font generates from the computer is different from real VLP image.
- The image generate is too clean, the noise level is zero, which is different from our test dataset

Figure4.4.6-2Synthesis Dataset VS Real Dataset

4.4.7 Data loading and Training

As the volume of our training data increasing, the amount of memory needed to load all the image into memory become impossible. This causes the whole training to kill at the beginning or middle of the training even if we increase the CPU memory to 64gb. Instead of load all the image and array into memory directory, we tried to store all the image arrays and labels into HDF5 format and load by batches. However, the train model doesn't converge during training, the image arrays and label retrieve is incompatible.



	0	1	2	3	4	5	6	7	8	9
0	1.0	1.0	1.0	1.0	33.0	9.0	3.0	1.0	1.0	30.0
1	1.0	1.0	1.0	33.0	26.0	21.0	10.0	4.0	10.0	1.0
2	1.0	1.0	1.0	33.0	11.0	6.0	1.0	3.0	3.0	12.0
3	1.0	1.0	1.0	33.0	31.0	17.0	9.0	4.0	4.0	3.0
4	1.0	1.0	1.0	1.0	33.0	7.0	10.0	10.0	4.0	21.0
5	1.0	1.0	1.0	33.0	27.0	18.0	4.0	4.0	8.0	10.0
6	1.0	1.0	1.0	12.0	23.0	31.0	2.0	5.0	6.0	8.0
7	1.0	1.0	1.0	33.0	24.0	28.0	3.0	7.0	2.0	9.0
8	1.0	1.0	1.0	1.0	33.0	24.0	34.0	10.0	7.0	4.0
9	1.0	1.0	1.0	33.0	20.0	13.0	10.0	8.0	3.0	2.0
10	1.0	1.0	1.0	33.0	33.0	31.0	2.0	5.0	7.0	9.0
11	1.0	1.0	1.0	12.0	16.0	33.0	3.0	8.0	9.0	2.0
12	1.0	1.0	1.0	33.0	20.0	13.0	8.0	4.0	5.0	3.0
13	1.0	1.0	1.0	33.0	28.0	16.0	10.0	10.0	8.0	8.0
14	1.0	1.0	1.0	33.0	29.0	21.0	10.0	6.0	10.0	1.0
15	1.0	1.0	1.0	1.0	33.0	27.0	29.0	8.0	7.0	4.0
16	1.0	1.0	1.0	1.0	12.0	14.0	35.0	7.0	1.0	10.0
17	1.0	1.0	1.0	33.0	21.0	22.0	8.0	9.0	2.0	8.0
18	1.0	1.0	1.0	12.0	16.0	11.0	8.0	6.0	9.0	5.0
19	1.0	1.0	1.0	33.0	35.0	11.0	10.0	4.0	2.0	9.0
20	1.0	1.0	1.0	33.0	35.0	31.0	3.0	2.0	6.0	1.0
21	1.0	1.0	1.0	1.0	33.0	2.0	5.0	5.0	2.0	29.0
22	1.0	1.0	1.0	33.0	11.0	2.0	2.0	1.0	3.0	15.0
23	1.0	1.0	1.0	33.0	11.0	5.0	4.0	6.0	4.0	33.0
24	1.0	1.0	1.0	18.0	12.0	11.0	9.0	10.0	1.0	9.0
25	1.0	1.0	1.0	1.0	33.0	30.0	26.0	3.0	7.0	6.0
26	1.0	1.0	1.0	33.0	27.0	27.0	10.0	6.0	4.0	9.0
27	1.0	1.0	1.0	12.0	18.0	32.0	6.0	2.0	6.0	6.0
28	1.0	1.0	1.0	33.0	27.0	22.0	4.0	1.0	4.0	8.0

Figure 4.4.5-1 HDF5 view of data

Finally, we decided to load the data during the training and by batches. However, this method is memory expensive, because the data load every in every epoch during the training.

4.4.8 Model Limitation

There are few limitations in our model, which includes,

- Bounded Sequence

  Due to the constraint of end-to-end sequence, where the output sequence must be bounded, our model is bounded with maximum 10 characters. In this circumstance, our model only able to make a prediction with maximum up to 10 characters.

- Unseen Pattern

  One of the disadvantages of Convolutional Neural Network is, CNN only able to predict features learned. In the case of feeding the model with the unseen pattern for prediction, our model will perform poorly. However, these unseen data, can continue to train on top of a trained model in order to support for the specific pattern of a dataset.

- Prediction Time

  The time took to make a prediction in both VGGNet15 or VGGNet19 on RNN or LSTM is about 7 to 8 seconds, which is quite lengthy for practical usage. However, this issue can be ignored with the advancement in computer processing speed.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this study, we have presented a license plate recognition system which adopted the end-to-end approach using deep learning. We designed and compared the recognition accuracy of CRNN model and CNN-LSTM model. We investigate through experiment by increasing the depth of the network using 2 different depth network, namely VGGNet15 and VGGNet19, the features learning abilities of the model would increase as well. Also, we compare the relationship between the depth of the network to the amount of data and the recognition performance through data augmentation techniques.

Our approach combines the advantages of feature learning and sequence labelling by joining both techniques, CNN, and RNN or LSTM together. Our experiment results demonstrate we able to achieve comparable performance to the manually engineered methods by which ours require no single pre-processing or post-processing and the needs for character level segmentation is completely avoided. Our approach allows the captured license plate to be pass and recognised in one forward pass and achieve accuracy up to 98.04% in LPR44, 94.84% in LPR45, and 65.96% for Open Environment Dataset. Besides, we also prove the VGGNet19 in our case, deeper architecture network able to learn more discriminate feature which is robust to various illumination, rotation and distortions in the image, and lead to higher recognition accuracy.

However, for deeper network architecture, sufficient training sample is a must in order to prevent overfitting. The low number of training sample in deep network architecture will cause the model to memorise the feature instead of generalising it. Also, efficiency is the question mark in deep network model, time taken for recognising each and every image took about average 7 to 8 seconds.

5.1 Future Work

5.1.1 Spatial Transformer Network

Spatial Transformer Network itself is a module. The basic idea is this module giving neural network ability to actively spatially transform feature maps. (Max, et al., 2016) The best of this module is able to make the model to be invariance to translation, scale, rotation which in traditional CNNs, we will need a lot of training samples that translated, rotated, and scaled. Also, it can be inserted into the existing Convolutional architectures without the need of any extra training supervision or modification to the optimization process. The advantages include resource and cost optimization as a single image is not augmented into translate, rotate or scale version and train at once.

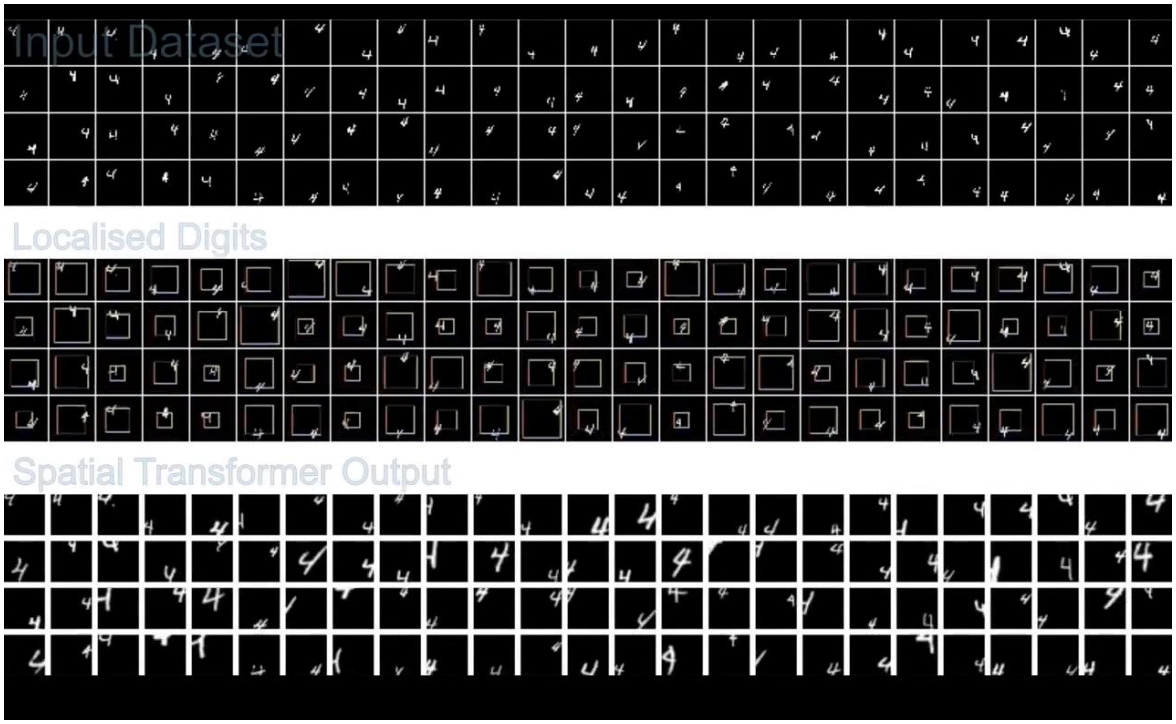


Figure 5.1.1-1Colocation Optimization Process Visualization- Initial Stage (Spatial Transformer Network, 20016)

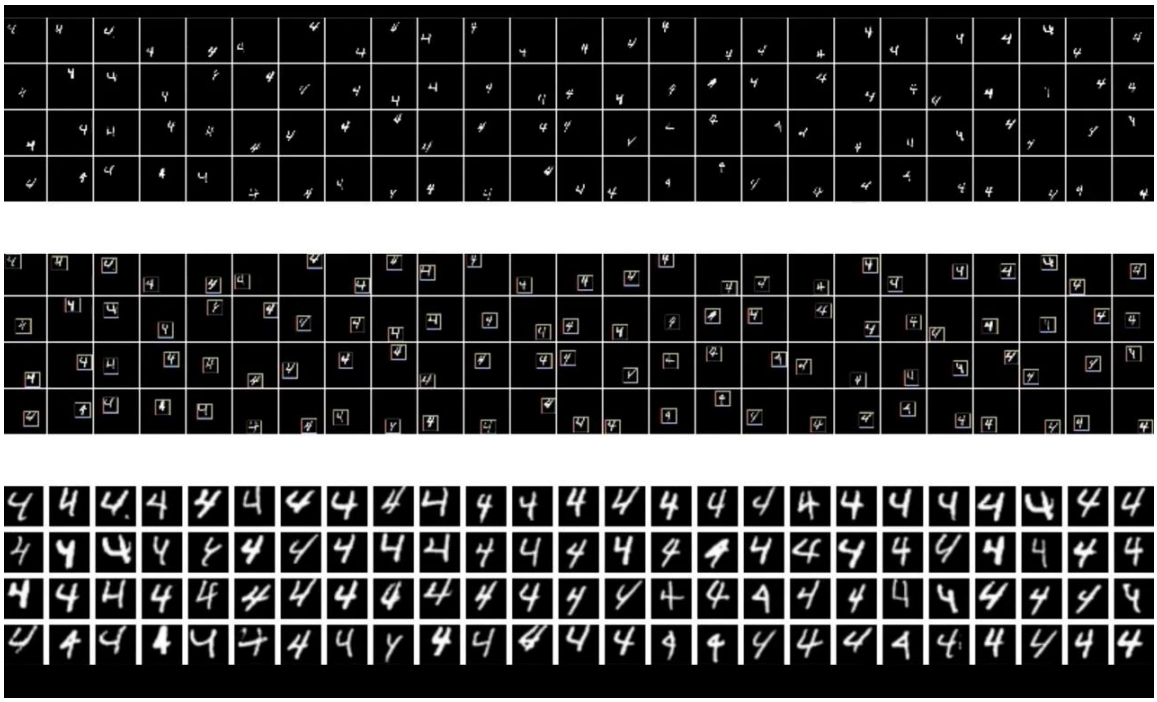


Figure 5.1.1-2Colocation Optimization Process Visualization - Final Stage(Spatial Transformer Network, 20016)

5.1.2 Dataset Expansion

The license plate recognizer performed badly on certain number pattern of the license plate. For example, Sabah, Sarawak, Kelantan, Pahang and Terengganu. The reason is because most of the training dataset having a pattern of Wilayah Persekutuan and Selangor license plate number. We should collect more license plate number pattern to ensure that data balancing. The reason for this is because convolutional neural network only can recognize the pattern that it has seen before.

5.1.3 Data Augmentation Techniques

In this paper, the augmentation techniques that we have apply are random rotation, random scaling, random translation and random blur sharpen effect. Other augmentation techniques should be added in order to improve the model's generalizability, for instance, noise augmentation includes salt and pepper noise, elastic deformations and etc.

REFERENCES

Anuja, P. N., 2011. License Plate Character Recognition System using Neural Network. *International Journal of Computer Application,* 25(10), pp. 36-39.

Bharat, B., Singh, S. & Ruchi, S., 2013. License Plate Recognition System using Neural Netowork and Multithresolding Technique. *International Journal of Computer Applications(0975-8887),* 84(5), pp. 45 - 50.

Bo, L., Bin, T. & Ye, L., 2013. Component-based license plate detection using conditional random field model. *IEEE Transactions on Intellignet Transportation Systems,* 14(4), pp. 1690-1699.

Chang, S., Chen, L., Chung, Y. & Chen, S., 2004. Automatic License Plate Recognition. *IEEE Transaction Intelligent Transportation System,* 5(1), p. 4253.

David, A. & Meng, X.-L., 2001. The Art of Data Augmentation. *Journal of Computational and Graphical Statistics,* 10(1), pp. 1-50.

Choo, K., Kueh, C., Chung, Y. and A. Suandi, S. (2012). Malaysian Car Number Plate Detection and Recognition System. Australian Journal of Basic and Applied Sciences, [online] 6(3), pp.49-59. Available at: http://ajbasweb.com/old/ajbas/2012/March/49-59.pdf [Accessed 24 Mar. 2016].

David, J. R. & Meghann, C., 2014. *Automated License Plate Recognition Systems:Policy and Operational Guidance for Law Enforcement,* Washington: s.n.

Du, S., Ibrahim, M., Shehata, M. & Badawy, W., 2013. Automatic License Plate Recognition(ALPR): A state of the art Review. *IEEE Transactions on Circuits and Systems for Video Technology,* 23(2), pp. 311-325.

Goel, S. & Dabas, S., 2013. Vehicle Registration Plate Recognition System using template matching. *International Conference on Advances in Signal Processing and Communication,* pp. 315-318.

Goodfellow, I. J. et al., 20147. *Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Network.* s.l., arXiv:1312.6082v4.

Guo, J. & Liu, Y., 2008. License plate localization and Character segementation with feedback self-learning and hybrid binarization techniques. *IEEE Transport Vehicle Technology,* 57(3), pp. 1447-1424.

Hui, L. & Shen, C., 2016. *Reading Car License Plates Using Deep Convolutional Neural Networks and LSTMs* Australia:arXiv, pp. 1-17

Ioffe, S. & Szegedy, C., 2015. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.* USA, arXiv:1502:03167, pp. 1-11.

Jeff, D. et al., 2015. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. s.l., s.n., pp. 1-14.

Karen, S. & Andrew, Z., 2015. Very Deep Convolutional Network For Large Scale Image Recognition. *ICLR 2015*. Hilton San Diego Resort & Spa, 7-9 May, 2015.

Karpathy, A., Li, F. & Johnson, J., n.d. *CS231n: Convolutional Neural Networks for Visual Recognition.* [Online] Available at: http://cs231n.github.io/convolutional-networks [Accessed 23 Mar 2016].

Liu, P., Li, G. & Tu, D., 2015. Low quality License Plate Character Recognition based on CNN.*2015 8th International Symposium on Computational Intelligence and Design.* HangZhou, 12-13 Dec, 2015. China:Curran Associates, pp. 53-58.

Liu, Y. & Huang, H., 2015. Car Plate Character Recognition using a Convolutional Neural Network with Shared Hidden Layers. s.l., s.n. pp. 638-643

Max, J., Karen, S., Andrew, Z. & Koray, K., 2016. *Spatial Transformer Network.* London, arXiv.

Rasheed, S., Naeem, A. & Ishaq, O., 2012. *Automated number plate recognition using hough lines and tempalte matching.* San Franscisco, WCECS 2012, pp. 199-203.

Richard, G. & Eric, L., 1996. A Survey of Methods and Strategies in Character Segmentation. *IEEE Transaction on Pattern Analysis and Machine Intelligence,* 18(7), pp. 690-706.

Schenkel, M. et al., 1994. *Recognition-based Segmentation of on-line Hand-Printed Words: input vs output segmentation. Pattern Recognition 27,* Holmdel, NJ 07733: AT&T Bell Laboratories.

Sneha, G. P., 2013. Vehicle License Plate Recognition using morphology and neural network. *Internatinal Journal on Cybernetics and Informatics,* 2(1), pp. 1-7.

Spatial Transformer Network. 2016. [Film] Directed by Jaderberg Max, Simonyan Karen, Zisserman Andrew, Kavakcuoglu Koray. London, UK: arxivSTmovie, Google DeepMind.

Zheng, L., He, X., Samali, B. & Yang, L., 2013. An algorithm for accuracy enhancement of license palte recognition. *Journal of Computer and System Science,* 79(2), pp. 245-255.