2016 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS2016)
17-20 December 2016, Tokyo Japan

Character Recognition of Malaysian Vehicle License Plate with Deep Convolutional Neural Networks

Dickson Neoh Tze How*, Khairul Salleh Mohamed Sahari†
Center for Advanced Mechatronics and Robotics (CAMaRo),
Universiti Tenaga Nasional,
Jalan IKRAM-UNITEN, 43000 Kajang, Selangor, Malaysia.
Email: *dickson@uniten.edu.my, †khairuls@uniten.edu.my

***Abstract*—This paper presents a vehicle license plate recognition method using deep convolutional neural networks. The focus of this paper is placed on the recognition of segmented characters of vehicle license. The deep convolutional neural network is able to distinguish numbers (0 to 9), alphabets (A to Z) and background image from one another. We show that the neural networks trained on computer fonts and natural images can be used to recognize the characters and non-characters on the vehicle license plates. In our experiments, we compared several models of the deep learning model and measure the performance of each model. We find that deeper models of neural networks yield better recognition results. What also find that the deep convolutional neural network is much more robust at the task of character recognition compared to the deep multilayer perceptron. With approximately equal amount of weights and biases parameters, the deep convolutional neural network outperforms all other models on the same task. Our best model using deep convolutional network, can achieve 95.89% correct classification of real license plate characters when even though the network is only trained on computer fonts (from Chars74K dataset) and natural images (from CIFAR10 dataset). No data augmentation is performed during the training.**

I. Introduction

Automatic license plate recognition (ALPR) is a surveillance method which utilizes optical character recognition to automatically recognize vehicle license plate characters. ALPR can be used in various areas to reduce the number of human supervision of the tasks. Tasks include unattended parking lots, security control of restricted areas, automatic toll collection [1], may find ALPR to be extremely useful. As of the current time, there are other robust ALPR libraries that documents good results such as the OpenALPR. Unfortunately, these libraries are tailor made to run on the US and European license plate which is quite different from the specifications of the Malaysian license plate. The current standard of Malaysian license plates are variations from the current plates from the United Kingdom. The general guidelines for any Malaysian vehicle plate requires the characters to be white and a black background for both front and rear plates regardless of the type of vehicles. The preferred typeface is typically Arial Bold and Franklin Gothic Bold. Detailed specifications can be obtained from the official portal of Road Transport Department of Malaysia. Typically ALPR consists of three main stages [1]. The first stage is locating the vehicle license plate, the second stage is segmentation of the plate characters and the third stage is recognizing the character from the segmented image. The focus of this study will be heavily placed on the third stage where we attempt to propose a deep learning based algorithm to recognize license plate characters. In subsequent sections, we describe our methods and also compare few neural networks based architectures and present the performance of each algorithm.

II. Methodology

A. Dataset Preparation

In this study we utilize a readily available dataset *Chars74K* which consists of characters typeset using computer fonts[2]. Another reason to using the *Chars74K* is that there are no openly available dataset of Malaysian vehicle license plate characters.

The Malaysian vehicle license plate only consists of white characters on black background, therefore, we removed colored characters from the Chars74K dataset. We also re-sized all the characters into 28 by 28 pixels. Figure 1 shows a few samples characters from the Chars74K dataset.

Fig. 1: Some of the character from the Chars74K dataset.

In total, the pruned Chars74K dataset consists of 35 distinct classes. These include numbers '0' to '9' and alphabets 'A' to 'Z' excluding the letter 'I' and 'O' and one additional class for classifying images that are not numbers or alphabets. The letter 'I' and 'O' is excluded because the Malaysian license plate does not contain these characters. Each class contains a sample 1016 images. We further divided the samples into training and validation sets for each class. Training 850 images out of 1016 is allocated as the training set and the remaining is allocated as the validation set. The division of the dataset into training and validation set allows us to implement the *early stopping* algorithm which will be described in the upcoming subsection. Apart from that, we have also collected a total of 730 cropped, real license plate characters from Malaysian vehicles. These

978-1-5090-6084-9/16/$31.00 ©2016 IEEE

images will be use to evaluate how well does the neural network generalize from training only using computer fonts. We address these actual pre-segmented characters as the *test set* in this study.

B. Network Topology

In this study, we utilized four different neural network models to benchmark the performance on the datasets. All four network varies in type, architectures, and number of parameters and hyperparameters involved. Table I tabulates the type and architecture of networks we utilize in this study. We address the each network by a name (Shown under the *Network name* column in Table I) .

TABLE I: A table of network types.

Network name	Network Type	Number of Layers	Total parameters
MLP-3	Multilayer Perceptron	3	682,531
MLP-6	Multilayer Perceptron	6	652,963
CNN-4	Convolutional Neural Network	4	604,035
CNN-6	Convolutional Neural Network	6	336,387

The MLP-3 model is a simple conventional feedforward multilayer perceptron (MLP). Since there is only one hidden layer, recent studies classified this type of network as a *'shallow network'*. The MLP-3 consist of a single hidden layer with 512 hidden units. All hidden units are configured to use the rectified linear activation function (or also known as Rectified Linear Units (ReLU)). All layers in the MLP-3 is densely connection to each other. This network architecture is chosen as the benchmark network due to its convention applications to images in the literatures[3].

The second network model is the slightly modified from the MLP-3. Instead of having only one hidden layer, the second network model contains four hidden layers. Two hidden layers contains 256 ReLU units, the remaining two hidden layers contains 128 ReLU units. All the layers in the MLP-6 model is densely connected to each other. Recent advances classified this type of network architecture as deep neural networks [**?**] and obtain good results on image classification tasks. We address this model as the MLP-6.

Recent advances in image classification tasks that obtain best results often include convolutional neural networks [4]. Since our study directly involves image classification tasks, we included the convolutional networks as network models for comparison with the conventional MLP. The third network model is a convolutional neural networks with 4 layers of computation. The architectures and sequence of layers are somewhat similar to the VGG convolutional neural networks by Simonyan *et. al.* [5] which obtained one of the best results in recent image classification tasks. There are three types of layers used in our implementation to the VGG-like convolutional neural network. The first is the convolution layer, ($CONV$). The convolutional layer convolve the input images with a filter and results in a feature map. Convolutional layers are the main element in the convolutional neural network architecture. The second type of layer is the pooling layer ($POOL$). Pooling layer performs downsampling of the data along a spatial dimension and results in a reduction of dimension size of the data. Recent state of the art convolutional neural networks often incorporate pooling layers as a vital layer to achieve improved results in image related tasks [6]. In this study all pooling layers performs the max pooling operation. The third layer type is the fully connected layer (FC). This units layer is a densely connected to all activation in the previous layers. The sequence of layers in the CNN-4 model is as follows $[INPUT- > CONV- > CONV- > POOL- > FC- > FC]$. Figure 2 illustrates the layers, connections and feature maps dimensions of the CNN-4. Similarly in our fourth model, CNN-6 we slightly modify the layer patterns of CNN model by adding more computational layers. CNN-6 has the following layer patterns $[INPUT- > CONV- > CONV- > POOL- > CONV- > CONV- > POOL- > FC- > FC]$. In total it has over 6 layers of computations. Figure 3 illustrates the layers, connections and feature maps dimensions of the CNN-6. All convolutional layers has a 3 by 3 filter size and all pooling layers has 2 by 2 size.

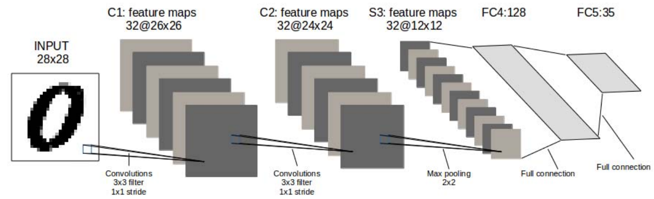


Fig. 2: The CNN-4 model.

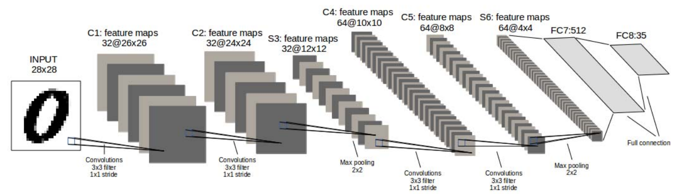


Fig. 3: The CNN-6 model.

C. Training

We utilized similar training procedure for all network models. The networks are trained using backpropagation with mini-batch gradient descent and using the adadelta optimizer [7]. The size of the mini-batch is 128. We utilized the categorical cross-entropy loss function as we are dealing with multi-class classification task. In order to avoid overfitting, we included dropouts [8] and also trained by the early stopping [9] technique. Algorithm 1 illustrates our implementation of the early stopping in our training. Figure 4 shows the error rate on the training set for all the network as training progresses.

III. Results and Discussions

This section elaborates the results we obtained by applying the methodology described. We have tested four different neural networks architectures on the task of vehicle plate character recognition using data from computer fonts. We

Algorithm 1 Early Stopping Training Algorithm

while $count < 100$ **do**
  Train network
  if no improvement on validation error **then**
    $count \leftarrow count + 1$
  end if
  if improvement on validation error **then**
    $count \leftarrow 0$
  end if
end while

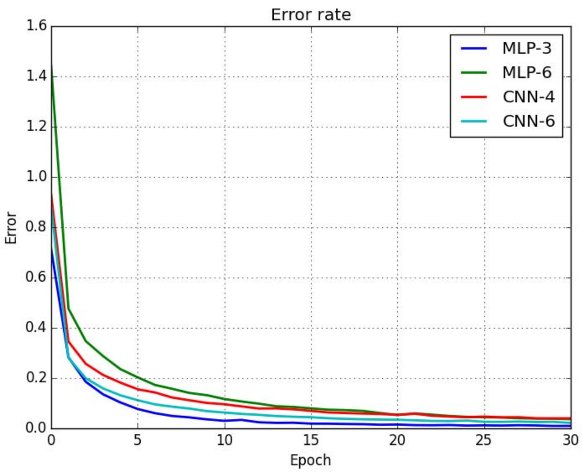


Fig. 4: Error rate of the models on the training set.

observe that even by using data from computer fonts without augmentation, the neural network model are able to perform reasonably well on the task. Table II tabulates the result on the recognition rate of the models. In Table II the *Network* column refers to the neural network model, *Layers* column show the number of layers the network has, *Parameters* shows the number of total weights and biases of the network, *Recognition rate(train)* shows the recognition rate on the Chars74K dataset and the *Recognition rate(test)* shows the recognition rate using real cropped Malaysia vehicle plate characters.

TABLE II: Recognition rate of five different models of deep neural networks on the training and test dataset.

Network	Layers	Parameters	Recognition rate (train)	Recognition rate (test)
MLP-3	3	682,531	99.97%	78.63%
MLP-6	6	652,963	99.89%	81.23%
CNN-4	4	604,035	99.69%	92.87%
CNN-6	6	336,387	99.91%	95.89%

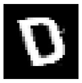
Based on the results in Table II the MLP-3 is the worst performer that scores the lowest accuracy on the test dataset despite scoring the best on the training dataset. This suggests that the MLP-3 is very prone to overfitting the training data. In MLP-6, we added three more layers to the MLP-3 while keeping the number of parameters at approximately the same and observed the recognition rate. The recognition rate on the test data improved by 2.6%. We find that adding more layers to the MLP improves its performance on the test data by reducing the tendency to overfit on the training data.

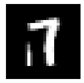
Adding on top of that, we also compared the performance of CNN to that of MLP. The CNN refers to a 4 layer convolutional neural network. The number of computation layer is the smaller than the MLP-6 model and the number of parameters are also slightly reduced. Having approximately similar parameters, the CNN boosts the recognition rate of the test data up to 11.55%. This huge performance boost verifies that the CNN architectures are very much suited to performing visual recognition tasks compared to the MLP.

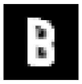
Since adding more layer to the MLP improves the performance of the network, we have also decided to all more depth to the CNN and monitored its recognition rate. The result is not out of expectation. The CNN-6, a 6-layer convolutional neural network, outperforms all other models. Its worth to note that the number of parameters of the CNN-6 is about half the size of all other network. Despite so, its recognition rate on the test data is still superior. This suggests that the CNN architecture is efficient in storing information in smaller amounts of weight parameters while maintaining recognition rate.

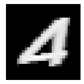
Figure 7 to Figure 10 illustrates the confusion matrix for all models on the test data images. We observe that the MLP tends to perform poorly on differentiating very similar characters. Some obvious mis-classifications can be seen in Figure 7. The MLP-3 confuses the number 7 and and number 1, number 8 for the letter H. In Figure 8 we see the same problem occurs but in lesser frequency as illustrated by the shades of the confusion matrix pixel. The problem is further alleviated by using the CNN as Figure 9 and Figure 10 shows.

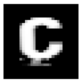
Figure 5 shows some of the misclassified characters using the CNN-6. Note the network tends to misclassify characters that are skewed in shape or slightly rotated in position. Since the Chard74K dataset does not contain the skewed or rototed images of computer fonts, the network does not generalize well on these types of characters. However, it is also surprising that the network can cope with most of the rotated and skewed characters. Many of the rotated and skewed characters were correctly classified by the CNN-6. Figure 6 shows some of the rotated and skewed characters that are correctly classified by the CNN-6.

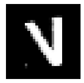

(a) Misclassified as 0.


(b) Misclassified as 1.


(c) Misclassified as 8.


(d) Misclassified as A.


(e) Misclassified as G.


(f) Misclassified as N.


(g) Misclassified as S.


(h) Misclassified as Z.

Fig. 5: Some of the misclassified characters by the deep convolutional network CNN-6.

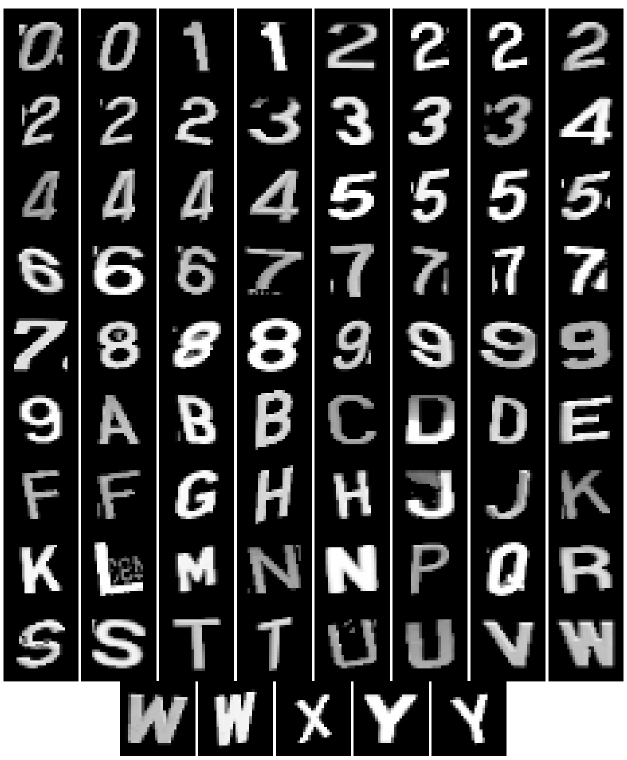


Fig. 6: Some of the correctly classified distorted characters from the test dataset.

IV. Conclusion and Future Works

In this work we have presented the performance comparison of four network architectures on the task of classifying the Malaysian vehicle pre-segmented license plate characters. We show that the networks can be trained on computer characters and can be made to recognize characters from real images of license plate characters with reasonable accuracy. We showed that even though the networks were only trained with computer characters without any distortions, skews and rotation the deep convolutional network is robust enough to correctly classify distorted images on the test set. In future improvements, we intend to augment the training dataset to include rotated, skewed and noisy images. Training with these expanded dataset has proven improve performance of the convolutional neural networks in many cases[4].

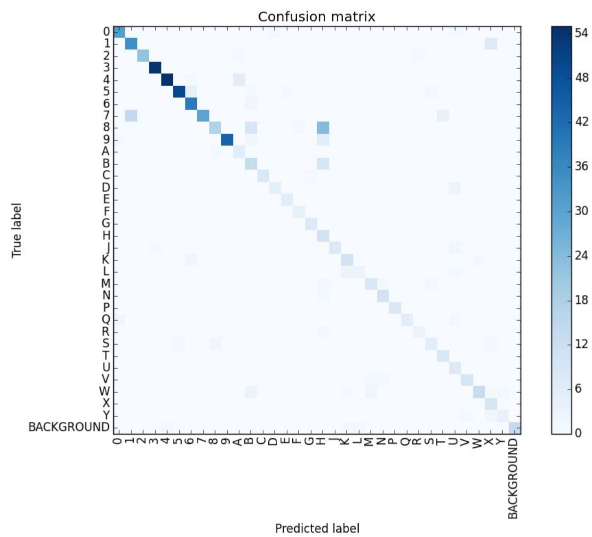


Fig. 7: Confusion matrix for the MLP-3.

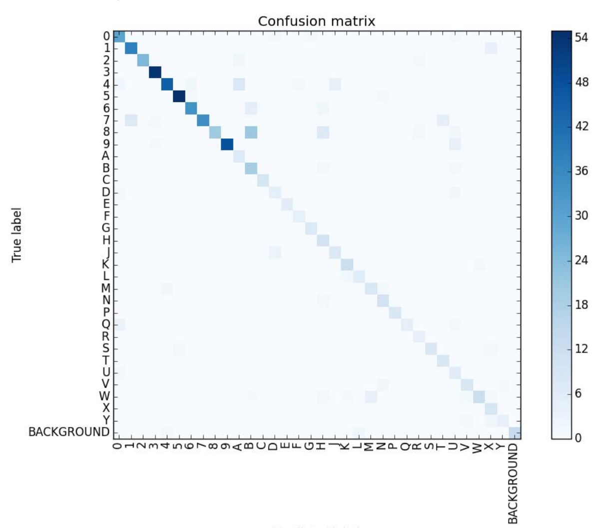


Fig. 8: Confusion matrix for the MLP-6.

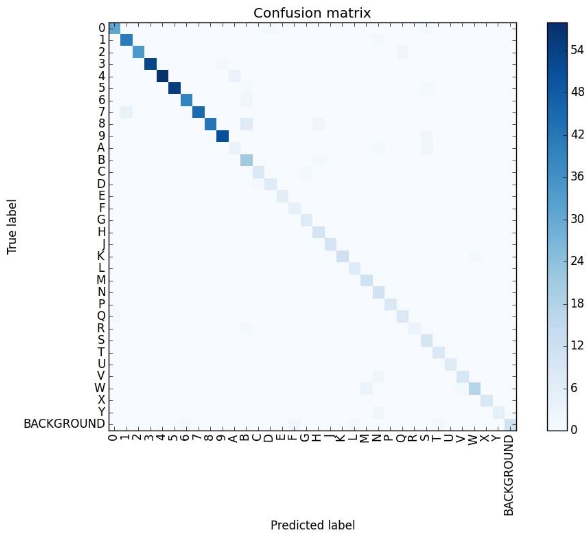


Fig. 9: Confusion matrix for the CNN-4.

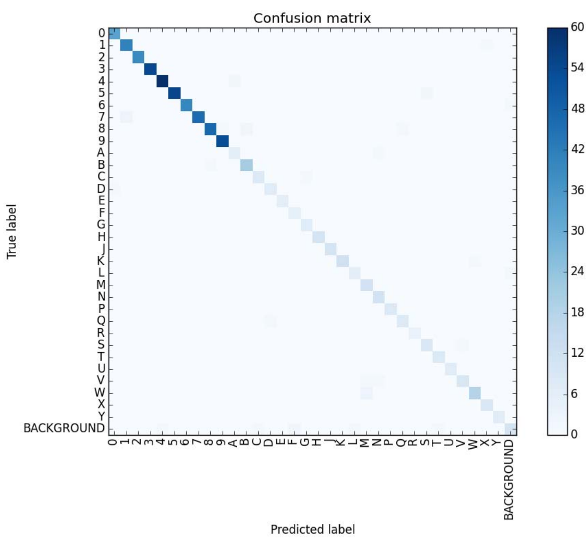


Fig. 10: Confusion matrix for the CNN-6.

References

[1] S.-L. Chang, L.-S. Chen, Y.-C. Chung, and S.-W. Chen, "Automatic license plate recognition," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 5, no. 1, pp. 42–53, 2004.

[2] T. E. de Campos, B. R. Babu, and M. Varma, "Character recognition in natural images." in *VISAPP (2)*, 2009, pp. 273–280.

[3] I. Kanellopoulos and G. Wilkinson, "Strategies and best practice for neural network image classification," *International Journal of Remote Sensing*, vol. 18, no. 4, pp. 711–725, 1997.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[6] J. Nagi, F. Ducatelle, G. Di Caro, D. Cireşan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, L. M. Gambardella *et al.*, "Max-pooling convolutional neural networks for vision-based hand gesture recognition," in *Signal and Image Processing Applications (ICSIPA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 342–347.

[7] M. D. Zeiler, "Adadelta: An adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[9] L. Prechelt, "Automatic early stopping using cross validation: quantifying the criteria," *Neural Networks*, vol. 11, no. 4, pp. 761–767, 1998.