Improved Optical Character Recognition with Deep Neural Network

Tan Chiang Wei
Intel Microelectronics (M) Sdn. Bhd.
Pulau Pinang, Malaysia
chiang.wei.tan@intel.com

U. U. Sheikh
Faculty of Electrical Engineering
Universiti Teknologi Malaysia
Johor Bahru, Malaysia
usman@fke.utm.my

Ab Al-Hadi Ab Rahman
Faculty of Electrical Engineering
Universiti Teknologi Malaysia
Johor Bahru, Malaysia
hadi@fke.utm.my

***Abstract*— Optical Character Recognition (OCR) plays an important role in the retrieval of information from pixel-based images to searchable and machine-editable text formats. In old or poorly printed documents, printed characters are typically broken and blurred, making character recognition potentially far more complex. In this work, deep neural network using Inception V3 is used to train and perform OCR. The Inception V3 network is trained with 53,342 noisy character images, which were collected from receipts and newspapers. Our experimental results show that the proposed deep neural network achieved significantly better recognition accuracy on poor quality text images and resulted in an overall 21.5% reduction in error rate compared to existing OCRs.**

***Index Terms*—OCR (Optical Character Recognition), Deep Learning, Transfer Learning**

I. Introduction

Text character recognition commonly deals with the recognition of optically processed characters and is also known as optical character recognition (OCR). The basic idea of OCR is to convert any hand written or printed text into data files that can be edited and read by machine. With OCR, any article or book can be scanned directly and the image can then be easily converted to text using a computer. The OCR system has two major advantages, which are the ability to increase productivity by reducing staff involvement and the ability to store text efficiently. Generally, the areas where OCR can be applied are postal departments, banks, publication industry, government agencies, education, finance and health care [1]. The universal OCR system consists of three main steps which are image acquisition and preprocessing, feature extraction and classification [1]. Image preprocessing phase cleans up and enhances the image by noise removal, correction, binarization, dilation, color adjustment and text segmentation. Feature extraction is to extract and capture information from the acquired text image to be used for classification. In the classification phase, the portion of the segmented text in the document image is mapped to the equivalent textual representation.

There are several existing OCR solutions which are commonly used in machine learning and pattern recognition. However, there is still a challenging problem for recognizing broken or faded English characters. The performance of OCR directly depends on the quality of input image or document, thus making character recognition in scene images is potentially far more complicated. In addition, poor quality English characters are typically obtained from old printed documents, and some are caused by damaged print cartridges. Unfortunately, these training samples are yet to be found in existing solutions. In order to recognize poor quality English characters, an improved OCR with sufficient training data is needed.

In transfer learning, training samples can be used to pre-train a network in the source domain, and these well-trained learning characteristics can be delivered and benefit the training process in the target domain of the second network. In recent years, traditional methods in the field of OCR research have been almost replaced with deep learning methods such as Convolutional Neural Networks (CNN). Oquab *et al.* proposed that using CNN to learn image representations on a large annotation dataset can adequately transfer this information to other visual recognition tasks with a limited amount of training data [2]. Yejun Tang *et al.* proposed to add an adaptation layer in CNN using transfer learning, which achieves performance improvement in historical Chinese character recognition tasks [3]. Inspired by these works, we propose to apply a deep neural network with transfer learning for broken English character recognition.

II. Methodology

A. OCR Design Model

The adopted methodology in this paper is inspired from YeJun Tang OCR system [3]. Although the research work done by YeJun Tang was focused on Chinese text characters, but the transfer learning concept can be applied for English text characters with the same purpose to reduce training time and improve recognition accuracy. A pre-trained model is used and applied together with transfer learning in this paper to enhance the recognition results and speed up the training process. The proposed OCR system has been designed with the help of various modules as shown in Fig. 1.

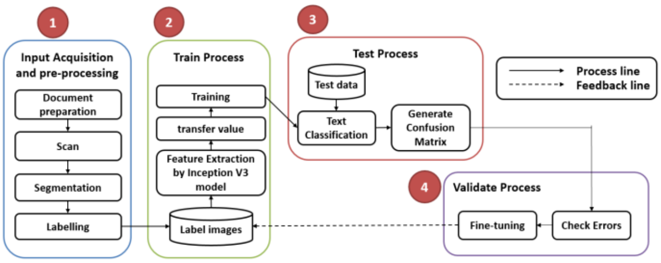


Fig. 1. The proposed OCR model

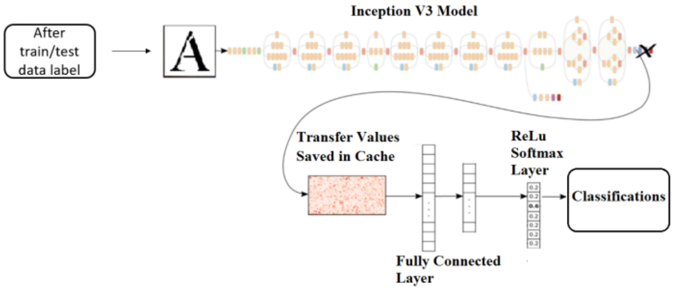


Fig. 2. Transfer-learning based on Inception V3 Model

The process is split into 4 major blocks (input acquisition and pre-processing, training, testing and validation). The first block is input acquisition and pre-processing of receipts and old newspapers. The prepared documents are scanned into images and these images will be processed and segmented using Maximally Stable Extremal Regions algorithm. After segmentation, all the segmented text characters become raw input that requires labeling. After labeling, the next step is to train using a deep neural network. In this work, the deep neural network that is used is a pre-trained Inception V3 model with transfer learning and is shown in Fig. 2. When all the labeled images have been processed through the Inception V3 model and the resulting transfer-values (weight values for classifier layers) saved to a cache file, then these transfer-values can be used as the input to another neural network. The second neural network is trained using the classes from the labeled images, so the network learns how to classify images based on the transfer-values from the Inception V3 model. In this way, the Inception V3 model is used to extract useful information from the images and another neural network is then used for the actual classification.

After 50,000 optimization iterations, the network is trained and ready for testing. Test dataset is given to the trained network for optical character recognition. Any fine-tuning in the validation stage requires a rerun of the network training and testing.

B. *Image Processing and Segmentation*

Image processing and segmentation is essential in extracting segments of symbols and text characters. The image processing strategy has four basic steps including grayscale conversion, binarization, dilation and segmentation. The process starts by converting image into grayscale image and then convert it to binary. Binarization helps to enhance the symbols and text characters and removes disjoint pixels. Characters are then enhanced by applying image dilation.

The next step is to perform character segmentation, which is to localize image blobs by applying a contour algorithm on the dilated image. Bounding Rectangle algorithm is used to bound each contour into the smallest possible rectangle. Fig. 3 shows the different segmentation results obtained for different kernel settings. When the kernel (x, y) size is set higher, then several characters are localized in the same rectangle box which is undesirable. A kernel of (3, 1) is found suitable in this work. Segmentation will generate text character images as output and these images are not yet labeled.

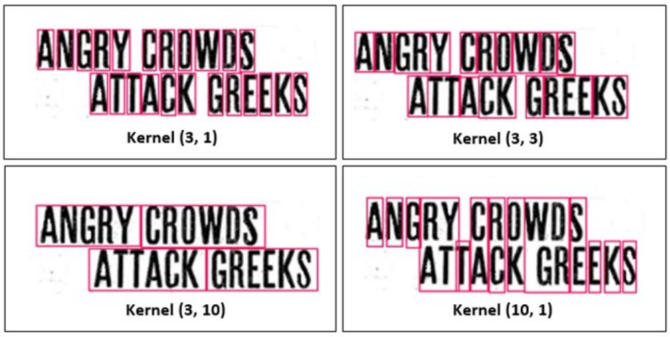

ANGRY CROWDS ATTACK GREEKS
Kernel (3, 1)
ANGRY CROWDS ATTACK GREEKS
Kernel (3, 3)
ANGRY CROWDS ATTACK GREEKS
Kernel (3, 10)
ANGRY CROWDS ATTACK GREEKS
Kernel (10, 1)

Fig. 3. Segmentation results using different kernel settings for bounding rectangle algorithm

III. Network Model Selection and Implementation

A. *Model Selection*

The network model consists of two networks, where the first network is for feature extraction and the second network is for classification. Three network models are trained and tested with text characters and the accuracy is shown in Table I.

TABLE I
First Network Model Selection

Network Model	Layers Details	Accuracy (%)	Training Time
10-layer CNN	5 conv/pool, 4 fully connected, 1 softmax	77.0	36 hours
5-layer CNN	2 conv/pool, 2 fully connected, 1 softmax	80.0	12 hours
Inception V3 Model	120 conv/pool, 2 fully connected, 1 softmax	90.6	1 hour 13 min

The first network model is a deep neural network with 10 layers of CNN and was initially used for flower recognition [4]. In this experiment, this network model is trained with a characters dataset (characters 0-9) and the accuracy obtained was 77%. The accuracy result is acceptable but the training time is too long for just 10 classes. The second network is a network model for CIFAR-10 dataset which is made up of 5 layers of CNN [5]. This network model is trained using CIFAR-10 dataset that consists of 60,000 color images with size of 32x32 for 10 classes (animals and transports). This dataset actually has 6,000 images per class and there are 50,000 training images and 10,000 test images. The accuracy obtained was 80%. The training process took around 12 hours and the result is acceptable for a network that just needs one time training. The last network is a pre-trained network model called Inception V3 model with transfer learning. In this experiment, this network model is trained with 94 classes of clean characters dataset (0-9, A-Z, a-z and symbols). Results show that using more layers can actually achieve higher accuracy of 90.6% compared to the other two network models. Besides that, the training time is just 1 hour 13 mins. Using a pre-trained model with transfer learning has the benefit of fast training time with high accuracy.

B. *Network Parameters*

The purpose of the following experiments is aimed at selecting best network parameters of the second network for classification.

1) Number of Fully Connected Layers: This experiment was done to analyze on how the accuracy and training time are affected by the number and size of the fully connected layers in the second network. When training a deep convolutional neural network, the first layer will train itself to recognize very basic things like edges, the next layer will train itself to recognize collections of edges such as shapes, and subsequent layers will learn more details. Table 2 shows that larger layer size results in longer training time. The highest accuracy obtained for a single fully connected layer is 64.5%, obtained not from the largest layer size of 4096 but was obtained from a layer size of 1024. Besides that, the highest accuracy obtained from two fully connected layers (1st: 4096, 2nd: 2048), with an accuracy of 66.6% but with a very long training time. Adding layers will also increase the training time, thus 3 fully connected layers is not considered.

2) Training Iterations: The next experiment was conducted to investigate on how the accuracy and time usage are affected by the number of training iterations. Table III illustrates that the accuracy increases at the beginning from 1,000 iterations until 50,000 iterations. The accuracy converges after 50,000 iterations and no improvement is noted.

TABLE II
EXPERIMENT ON THE NUMBER OF FULLY CONNECTED LAYERS

Number of Layers	Layer Size (1st, 2nd, 3rd)	Accuracy (%)	Time (mins)
1	256	63.0	11.72
	512	62.9	14.20
	1024	64.5	20.30
	2048	64.1	31.15
	4096	64.4	54.02
2	1024, 256	61.1	21.60
	2048, 256	61.9	35.23
	4096, 1024	65.8	75.61
	1024, 4096	65.6	44.28
	4096, 2048	66.6	105.05
3	4096, 2048, 1024	61.4	105.73
	4096, 1024, 256	61.6	75.32
	1024, 4096, 1024	60.9	63.40
	4096, 1024, 1024	64.4	81.88
	4096, 4096, 1024	61.0	158.2

TABLE III
ITERATION VS ACCURACY AND TIME USAGE

Iteration No	Accuracy %	Time (mins)
1000	40.40	1.50
5000	63.80	7.48
10000	63.80	18.32
50000	65.50	73.50
100000	63.20	146.13
150000	65.10	221.57
250000	65.70	470.62

3) Batch Size: Instead of testing the number of iterations, in this experiment, the batch size is evaluated. Batch size defines the number of image samples that are propagated through the network. Batch size does influence training time and accuracy but there is no general rule for determining the optimal batch size. We experimented with 3 different batch size; 32, 64, and 128. Table IV shows that increasing the batch size will increase the training time. Besides that, batch size of 64 obtained the highest accuracy and acceptable time. In order to avoid overfitting (such as when batch size is 128), batch normalization or dropout is applied. Table V shows that by using dropout equals 0.2 overcome overfitting (dropout 20% of batch size).

TABLE IV
BATCH SIZE VS ACCURACY

Batch Size	Time (mins)	Accuracy (%)
32	70.32	62.8
64	73.35	73.1
128	82.57	64.1

TABLE V
DROPOUT TEST ON BATCH SIZE 128

Batch Size	Dropout Rate	Batch Size after dropout (approx)	Time (mins)	Accuracy (%)
128	0.8	25.6	81.23	69.0
128	0.5	64.0	81.02	68.8
128	0.2	102.4	83.40	70.6

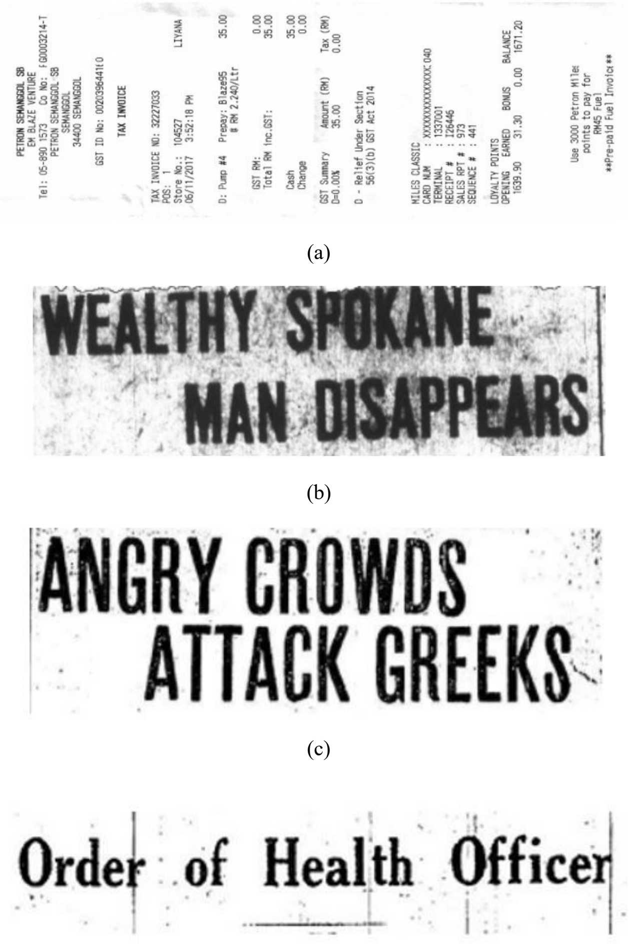
4) Activation Function: The purpose of activation function is to convert an input signal of a node in a neural network to an output signal [6]. Without activation function, neural networks would not be able to learn and model complicated information such as videos, audio, speech and images [7]. There are different types of activation function that can be applied such as Relu (Rectified Linear Unit), sigmoid and tanh. We have conducted accuracy measurement using ReLu, sigmoid and tanh. Experimental results show that ReLu has the highest accuracy (73.1%) while sigmoid and tanh achieved 67.5% and 68.8% accuracy respectively. ReLu has no gradient vanishing problem as ReLu's gradient is always constant = 1. The only disadvantage of ReLu is that it can cause overfitting, but this can be solved by using dropout. The final network implementation is as follows. We use the Inception V3 network model, with two fully connected layers (1st: 4096, 2nd: 2048). We train the final network with 50,000 iterations, a batch size of 128 with a dropout factor of 0.2 and ReLu is used as the activation function for the proposed OCR system.

IV. RESULTS

A. *Benchmark with Existing OCR*

The proposed OCR is benchmarked against existing OCR, the a9t9 (also called as OCR space). The a9t9 OCR was released in 2017 and the API is available online [8]. a9t9 OCR supports image dimensions of 40 by 40 pixels up to 2600 by 2600 pixels. Since a9t9 OCR does not provide any standard testing data in public, thus the collected testing-sets from real-world samples are used for benchmarking.

We use four real-world samples (see Fig. 4 (a-d)), containing 657 text characters to benchmark both OCR systems. Table VI indicates that the proposed OCR achieves better performance compared to a9t9 OCR when it comes to poor quality test samples.



PETRON SEMANGGOL SB
EM BLAZE VENTURE
Tel: 05-890 1573 Co No: FG0003214-T
PETRON SEMANGGOL SB
SEMANGGOL
34400 SEMANGGOL

GST ID No: 002039644160

TAX INVOICE

TAX INVOICE NO: 32227033
POS: 1
Store No.: 104527 LIYANA
06/11/2017 3:52:18 PM

D: Pump #4 Prepay: Blaze95 35.00
@ RM 2.240/Ltr

GST RM: 0.00
Total RM Inc.GST: 35.00

Cash 35.00
Change 0.00

GST Summary Amount (RM) Tax (RM)
D=0.00% 35.00 0.00

D - Relief Under Section
56(3)(b) GST Act 2014

MILES CLASSIC
CARD NUM : XXXXXXXXXXXXXXXX040
TERMINAL : 1337001
RECEIPT # : I26446
SALES RPT # : 973
SEQUENCE # : 441

LOYALTY POINTS
OPENING EARNED BONUS BALANCE
1639.90 31.30 0.00 1671.20

Use 3000 Petron Miles
points to pay for
RM45 Fuel
Pre-paid Fuel Invoice

(a)

WEALTHY SPOKANE
MAN DISAPPEARS

(b)

ANGRY CROWDS
ATTACK GREEKS

(c)

Order of Health Officer

(d)

Fig. 4. (a-d) Test samples used for benchmarking.

TABLE VI
BENCHMARK RESULTS BETWEEN A9T9 AND THE PROPOSED OCR

Test Image	Total Characters In Image	a9t9 OCR		Proposed OCR	
		Recognized	Accuracy (%)	Recognized	Accuracy (%)
(a)	587	254	43.3	409	69.0
(b)	27	18	66.7	19	70.4
(c)	23	6	26.1	19	82.6
(d)	20	18	90.0	19	95.0

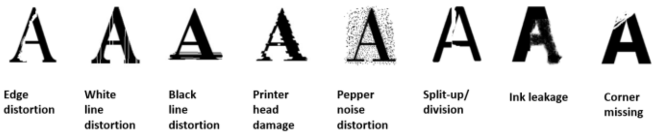


Fig. 5. Eight different noise patterns used in noise pattern analysis.

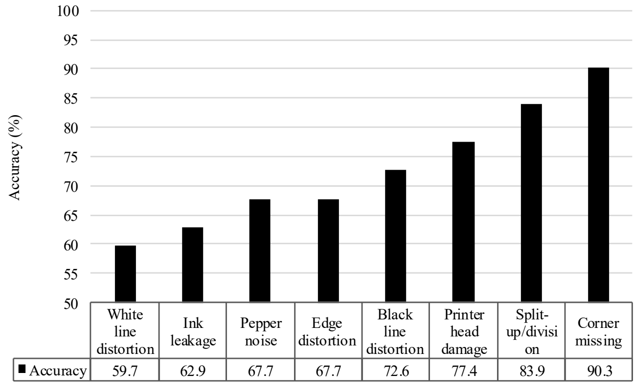


Fig. 6. OCR classification accuracy for different simulated noise patterns.

B. Text Character's Noise Pattern Analysis

The aim of this experiment is to analyze the impact of different noise patterns on the detectable rate of text characters. Fig. 5 shows the hand-crafted test data. The hand-crafted data is specially designed with 8 different noise patterns for each class, and all these hand-crafted data will be used in this experiment. A total of 135 images were used.

Fig. 6 shows that characters with "corner missing" and "split-up" noise patterns are easily classified compared to other types of noise. Notable, the proposed OCR performed poorly when subjected to characters distorted with "white line", "ink leakage" and "pepper noise". The recognition accuracy is greatly affected by the damaged area in the image. In order to improve the classification rate, the deep neural network must be trained with sample images of similar kind of noise patterns.

V. CONCLUSION

In this work, we proposed an OCR system for the recognition of printed text in poor quality images. This was achieved by building a transfer learning based OCR using a pre-trained deep neural network (Inception V3). The propsed deep neural network based OCR was trained and tested using real world samples and standard English Text Character dataset. From the experiment results, the system achieved significantly better recognition accuracy at an average of 78% for poor quality text images and resulted in overall 21.5% reduction in error rate as compared to a9t9 OCR. Furthermore, the OCR also maintained a 90.6% accuracy on good quality image test dataset. Further experiments conducted showed that the proposed OCR can perform well with noisy character images. The results also concluded that training process for a network must not only include good quality training data but also poor quality training data to improve the learning of the network.

Acknowledgments

The authors thank the Ministry of Education Malaysia and Universiti Teknologi Malaysia (UTM) for their support under grant number Q.J130000.2623.14J48.

References

[1] R. Anil, K. Manjusha, S. S. Kumar, and K. P. Soman, “Convolutional Neural Networks for the Recognition of Malayalam Characters,” in Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014, vol. 328, Springer International Publishing, 2015, pp. 493–500.

[2] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, “Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks,” in Proceedings of the 2014 Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1717–1724.

[3] Y. Tang, L. Peng, Q. Xu, Y. Wang, and A. Furuhata, “CNN Based Transfer Learning for Historical Chinese Character Recognition,” in Proceedings of the 12th IAPR Workshop on Document Analysis Systems (DAS), 2016, pp. 25–29.

[4] Cho Yee Phy, “Tensorflow input image by tfrecord,” 2017. [Online]. Available: https://github.com/yeephycho/tensorflow_input_image_by_tfrecord. [Accessed: 4-Mar-2018].

[5] L. Hvass, “Tensorflow tutorial 06 CIFAR-10,” 2016. [Online]. Available: https://www.youtube.com/ watch?v = 3BXfw_1_TF4. [Accessed: 4- Mar- 2018].

[6] R. Prajit, Z. Barret, and V. L. Quoc, “Searching for Activation Function,” arXiv, vol. 2, Oct. 2017.

[7] H. Chung, S. J. Lee, and J. G. Park, “Deep neural network using trainable activation functions,” in Neural Networks (IJCNN), 2016 International Joint Conference on, 2016, pp. 348–352.

[8] OCR SPACE, “Free OCR API and Online OCR.” 2018. [Online]. Available: https://ocr.space/. [Accessed: 4- Mar-2018].