Optical Character Recognition performance comparison of Convolution Neural Network and Tesseract

D. G. Ko[1], S. H. Song[1], K. M. Kang[1], S. W. Han[1] and J. H. Yi[2]
[1] Imaging Lab, Samsung Electronics
130, Samsung-ro, Yeongtong-gu, Suwon-si, Gyeonggi-do, Republic of Korea
[2] College of Information & Communication Engineering, Sungkyunkwan University
2066, Seobu-ro, Jangahn-gu, Suwon-si, Gyeonggi-do, Republic of Korea
E-mail: [1]{ dg.ko, suhan.song, kimin.kang, seoungwook.han }@samsung.com, [2]jhyi@skku.edu

Abstract: OCR (Optical Character Recognition) is text recognition which is designed to extract ASCII code from an image. It is one of the most important method in scanner application for TTS (Text To Speech) and automatic document classification. The OCR is still a challenging field in computer vision. One of the examples of OCR software packages is Tesseract that is made by HP. Especially, interesting part from our point of view is Deep Learning. Nowadays, Deep Learning has been used in many fields, such as classification, detection, tracking and recognition.

Our main idea is to compare two methods in OCR: one is convolution neural network (CNN) based training system; another is Tesseract based pattern recognition. In this paper, we used Caffe OCR character sets to measure recognition accuracy and processing time between CNN and Tesseract. As a result, we recommend a CNN due to its performance on recognition accuracy. However, if processing time is on priority, we recommend a Tesseract for its speed.

Keywords-- **OCR, convolution neural network, Deep-Learning, Tesseract**

1. Introduction

OCR (Optical Character Recognition) [1] is widely used scanner application, and it can recognize both printed text and handwritten text in an image. In addition, the OCR performance is directly dependent on quality of input image or document. So far, the OCR cannot be compared with human reading capabilities. Therefore, in an engineering aspect, the capability needs to be improved.

By the middle of the 1950's OCR machines are commercially launched. In 1960's to 1970's, the OCR system was able to recognize regular printed text and hand printed text. For the new version of an OCR, appeared in the middle of the 1970's, could recognize poor quality text and hand written characters. And nowadays, the OCR system is improved in its performance and starts to be provided as software package.

In recently years, MFP (Multi Function Printer) and high speed scanners are developed, and customers demand various applications, such as OCR, over-scan, automatic documents classification [2], skew correction [3], TTS (Text To Speech) [4] and ROI scan. The OCR is one of the most important tasks to solve document classification and skew correction.

Character recognition is a subset of pattern recognition area. However, it can approach Deep Learning based on training system. The goal of this paper is to compare two method in OCR between convolution neural network (CNN) [9,10,12,13,16] and Tesseract [6,7]. For a test, we used Caffe OCR character sets [8], these character sets with inserted noise (either Salt and Pepper noise or Gaussian noise) [Fig. 1]. The performance is measured by recognition accuracy and processing time.

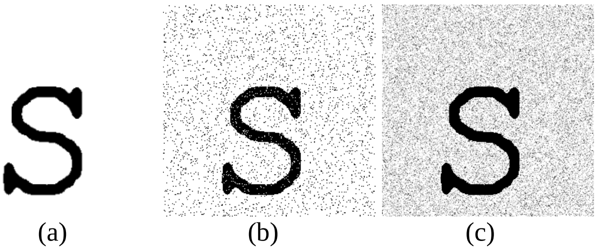

Figure 1. An example of character used in the experiment: (a) noiseless character (b) noisy character with Salt and Pepper (c) noisy character with Gaussian noise

The rest of the paper is organized as follows. In the next section, we talk about overview of OCR and delineate both CNN and Tesseract. In section 3, we will show a method designed both CNN and Tesseract for OCR. We also talk about experimental results with performance comparing by CNN and Tesseract. In section 4, we derive a conclusion about OCR method.

2. OCR methods

In this section, we describe OCR methods of CNN and Tesseract.

2. 1 CNN

Deep Learning has emerged as important area in AI (Artifact Intelligence), ML (Machine Learning), CV (Computer Vision), due to rapid development in digital image processing with huge and high quality datasets. The goal of Deep Learning method is to find a solution that best maps a set of correct output. The examples are handwritten text recognition [12], image classification [9,10] and object detection [11] tasks. Their methods are to approach focus on deep convolutional neural networks, and this method imrpoved by Y. LeCun *et al* [13]. In the work, a new framework for digits recognition method was proposed with LeNet-5 that comprise 7-layers using convolutional neural network.

CNN is perfectly validated by huge datasets as CIFAR-10/100 [14] and ILSVRC [15] (ImageNet Large-Scale

Table 1. List of character sets

Datasets	Image components	The number of character
Caffe OCR	Natural number 0 ~ 9, Upper case alphabet A ~ Z	6,400 characters for training 2,700 characters for test
VCOCR	Same as Caffe OCR	Same as Caffe OCR's test sets
NCOCR	Same as Caffe OCR	2,700 characters randomly selected from Caffe OCR with either salt and pepper noise or gaussian noise inserted

Visual Recognition Challenge). They showed strong correction about the image classification. Howerver, to training them requires much time on CPU process. Nowadays, GPU has overcome this problem with fast parallel processing.

OCR using CNN case is introduced by *pannous* [16]. Our approach for CNN is similar to LeNet-5 designed by Y. LeCun *et al*. Input image is 256 by 256 pixel and gray image. In addition, it is consists of 7 layers, 36 labels with '0' to '9' and 'A' to 'Z'. Lastly, we used the 6,400 characters from Caffe OCR character sets for training as shown in Table 1.

2. 2 Tesseract

Tesseract is an Open Source for the OCR engine that was developed by HP between 1984 to 1994. The engine was sent to UNLV for Annual Test of OCR Accuracy in 1995. In 2005, Tesseract was released as Open Source by HP [19].

Tesseract had independently developed page layout analysis technology. Therefore, Tesseract assume that their input images are a binary image that can handle both positive text (white on black text) and negative text (black on white text). Tesseract is structured in the black diagram as shown in Fig. 2. The procedure for Tesseract process is as follows:

1. Adaptive threshold: this step is to get a binary image from the examples of lightness non-uniformity image
2. Connected component analysis: connected components analysis and characters outline extraction in the binary image
3. Line & word finding: the outline are converted into Blobs
4. Word recognition: the result from step 3 is classified and the rest of the word recognition step applies only to non-fixed-pitch text.
5. Extracts the text from given an image: producing the output text

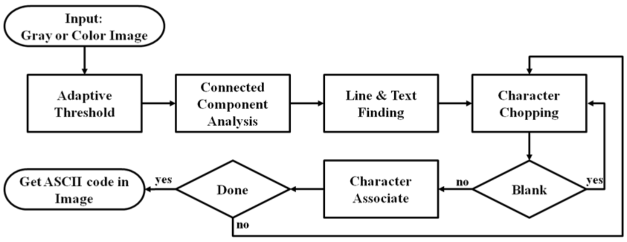

Figure 2. Architecture of Tesseract

3. Performance Comarison

CNN for OCR needs a training set with optimized DB (Data Base) and we used a Caffe OCR training character sets. It has several types of fonts, such as AndaleMono, Arial, ComicSansMS, CourierNew, Georgia, Impact, TimesNewRoman, TrebuchetMS and Verdana. In addition, we generated Caffe OCR character sets as Fig. 3.1 to Fig 3.2: white on black text to black on white text and resized 256 x 256 for general research. In a NCOCR (Noisy Caffe OCR) character sets case, the validated samples are selected randomly from Caffe OCR with random switching either Salt and Pepper noise or Gaussian noise inserted.

Figure 3.1. 28x28 white on black text original characters

Figure 3.2. 256x256 modified black on white text characters

We will show performance with recognition accuracy and processing time. Our testing environment of desktop is Microsoft Windows 7 64-bit, Intel Core i5-2320, RAM of 8 GB and using SSD.

3. 1 CNN

We trained using Caffe OCR training set, and it took about two weeks on training based on our CPU environment. We obtained accuracy rate on training by every 100 iteration times as follow graph [Fig. 4]. Moreover, we also obtained optimized-DB by every 100 iteration times. Using optimized-DB of every 100 iteration, we computed recognition accuracy on VCOCR [Fig. 5], and the graph is to show overfitting after 13,000 iteration. As on outcome of this tendency, we select a 9,000 iteration representing CNN experiment.

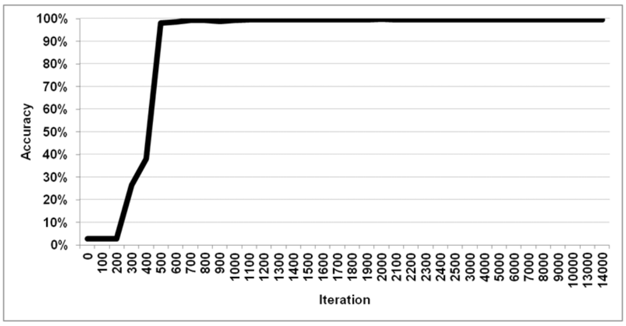

Figure 4. Accuracy in test by training step

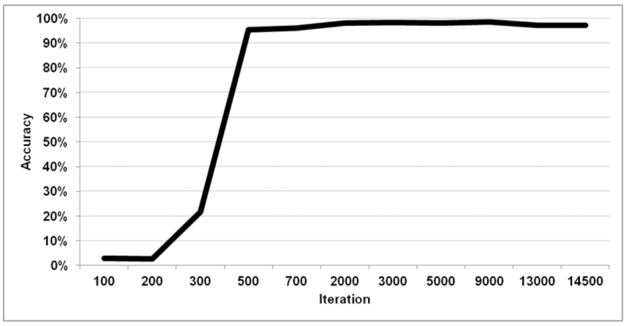

Figure 5. Test error by each iteration

The recognition accuracy of VCOCR character sets experiment was 98.67% [Table 2], and the number of misrecognition of each characters are shown as in Fig. 6 on VCOCR. Alphabet ‘Y’ is misrecognized alphabet ‘V’, number ‘0’ is recognized as ‘O’, the reverse is also appeared.

Table 2. Recognition result by VCOCR

	Success	Fail
The number of characters	**2664**	**36**
Probability	**98.67%**	**1.33%**

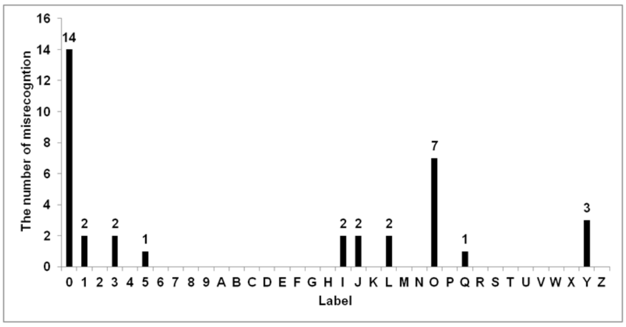

Figure 6. The number of misrecognition on VCOCR

The recognition rate is 96.52% on NCOCR [Table 3]. The tendency of recognition error of each character is not seen such as ‘L’ is misrecognized ‘U’, ‘D’, ‘J’ and ‘E’ [Fig. 7].

Table 3. Recognition result of NCOCR

	Success	Fail
The number of characters	**2606**	**94**
Probability	**96.52%**	**3.48%**

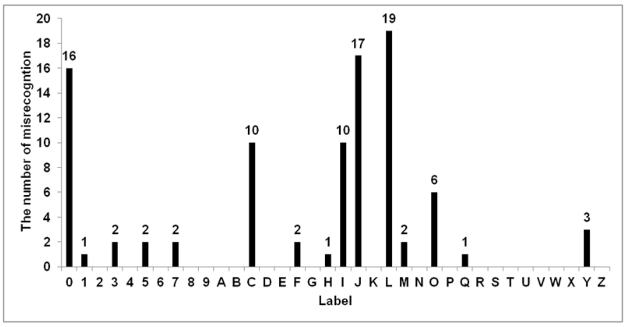

Figure 7. The number of misrecognition on NCOCR

Finally, processing time of CNN is 6,661 second.

3. 2 Tesseract

The result of VCOCR is shown in Table 4, and the number of misrecognition of each character is shown in Fig. 8.

Table 4. Recognition result of VCOCR

	Success	Fail
The number of characters	**2298**	**402**
Probability	**85.11%**	**14.89%**

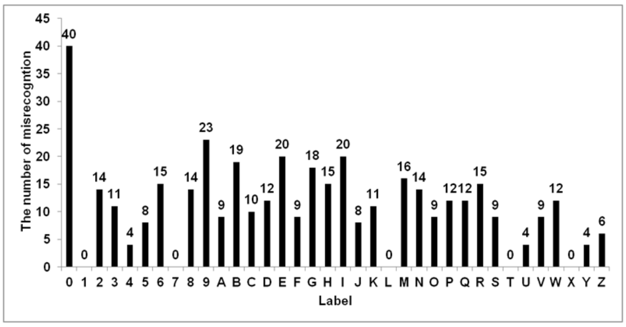

Figure 8. The number of misrecognition on VCOCR

In the NCOCR case, Tesseract is showed that has one great weakness about noisy [Table 5]. The number of misrecognition of each character is shown in Fig. 9.

Table 5. Recognition result of NCOCR

	Success	Fail
The number of characters	**1473**	**1227**
Probability	**54.56%**	**45.44%**

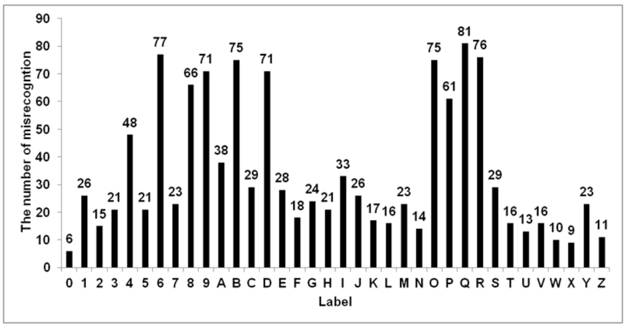

Figure 9. The number of misrecognition on NCOCR

Lastly, processing time of Tesseract is 803 second.

4. Conclusion

Our mainly approach is performance comparing an OCR: one method is CNN; another method is Tessearct. The result of recognition accuracy is to compare CNN with Tesseract as shown in Fig. 10. CNN is showed higher performance than Tesseract. However, In the processing time aspect, Tsseract is better than CNN [Table 6].

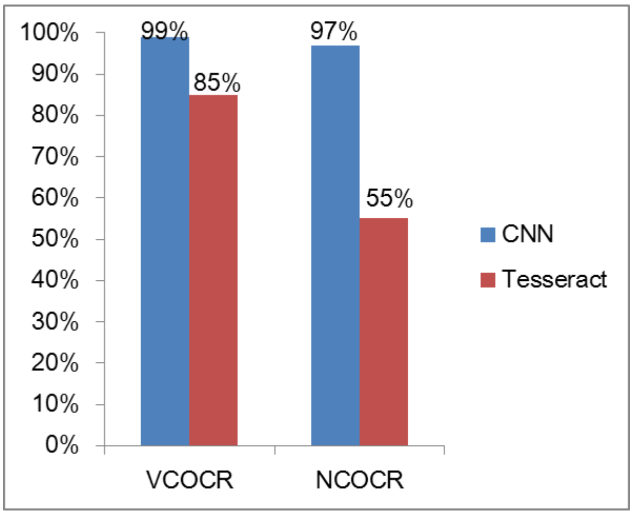

Figure 10. Experiment result into recognition accuracy

Table 6. Processing times

	Process Time
CNN	**6,661 sec**
Tesseract	**803 sec**

References

[1] R. Mither, S. Indalkar and N. Divekar, "Optical Character Recognition", *International Journal of Recent Technology & Engineering*, IJRTE, 2013.

[2] M. Dillgenti, P. Frasconi and M. Gori, "Hidden Tree Markov Models for Document Image Classification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, 2003.

[3] Y. Cao, S. Wang and H. Li, "Skew detection and correction in document images based on straight-line fitting", *Pattern Recognition Letters, ELSEVIER*, 2003.

[4] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis", *Proceedings of the 1998 IEEE International Conference on*, IEEE, 1998.

[5] L. Shafarenko, M. Petrou and J. Kittler, "Automatic watershed segmentation of tandomly textured color images", *IEEE Transactions on Image Processing*, IEEE, 1997.

[6] R. Smith, "Tesseract OCR Engine", Google Inc, OSCON, 2007.

[7] R. Smith, "An Overview of the Tesseract OCR Engine", *International Conference on Document Analysis and Recognition*, IEEE, 2007.

[8] C. Patel, A. Patel and D. Patel, "Optical Character Recognition by OpenSource OCR Tool Tesseract: A Case Study", *International Journal of Computer Applications*, 2012.

[9] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", *Advanced in Neural Information Processing System 25*, NIPS, 2012.

[10] D. Ciresan, U. Meier and J. Schmidhuber, "Multi-column Deep Neural Networks for Image Classification", *Computer Vision and Pattern Recognition*, IEEE, 2012.

[11] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", *Computer Vision and Pattern Recognition*, IEEE, 2014.

[12] D. Ciresan, U. Meier, L. M. Gambardella and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition", *Neural Computation*, MIT Press Journals, 2010.

[13] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient based learning applied to document recognition", *Proceedings of the IEEE*, IEEE, 1998.

[14] A. Krizhevsky, "Convolutional Deep Belief Networks on CIFAR-10", *Unpublished*, 2010.

[15] J. Deng, A. Berg, S. Satheesh, H. Su and A. Khosla, "Large scale visual recognition challenge 2012", *ImageNet*, 2012.

[16] Caffe OCR, https://github.com/pannous/caffe-ocr, available.

[17] BLVC (Berkeley Vision and Learning Center), caffe.berkeleyvision.org, *Caffe Deep Learning framework*, available.

[18] F. Shafait, D. Keysers and T. M. Breuel, "Efficient Implementation of Local Adaptive Thresdholding Techniques Using Integral Images", *Annual Symposium on Electronic Imaging*, IS&T/SPIE, 2008.

[19] Tesseract, http://code.google.com/p/tesseract-ocr, available.