IEIE Transactions on Smart Processing and Computing, vol. 6, no. 1, February 2017
https://doi.org/10.5573/IEIESPC.2017.6.1.053

Convolutional Neural Networks for Character-level Classification

Dae-Gun Ko, Su-Han Song, Ki-Min Kang, and Seong-Wook Han

Imaging Lab, Samsung S-Printing Solution Co., LTD. / Suwon, Korea
{dg.ko, suhan.song, kimin.kang, seongwook.han}@samsung.com

***** Corresponding Author: Dae-Gun Ko

Received December 26, 2016; Accepted January 11, 2017; Published February 28, 2017

** Regular Paper*

***Abstract*:** Optical character recognition (OCR) automatically recognizes text in an image. OCR is still a challenging problem in computer vision. A successful solution to OCR has important device applications, such as text-to-speech conversion and automatic document classification. In this work, we analyze character recognition performance using the current state-of-the-art deep-learning structures. One is the AlexNet structure, another is the LeNet structure, and the other one is the SPNet structure. For this, we have built our own dataset that contains digits and upper- and lower-case characters. We experiment in the presence of salt-and-pepper noise or Gaussian noise, and report the performance comparison in terms of recognition error. Experimental results indicate by five-fold cross-validation that the SPNet structure (our approach) outperforms AlexNet and LeNet in recognition error.

***Keywords*:** OCR, Deep-Learning, Convolutional neural networks, MNIST

1. Introduction

Optical character recognition (OCR) [4] provides a solution to automatic recognition of printed text or handwritten text in an image. OCR has been widely used as a scanner application. However, the performance of OCR is directly dependent on the quality of the input image or document, and still cannot compare with human character recognition in the presence of noise. In order to improve OCR technologies, analysis of the current state-of-the-art OCR methods is pertinent.

OCR machines first appeared on the market in the mid-1950s. In the 1960s to the 1970s, OCR systems were able to recognize regular printed text and hand-printed text. A new version of OCR that appeared in the middle of the 1970s could recognize poor-quality text and hand-written characters. The performance of OCR systems has been continuously improving since then, and OCR solutions have been provided as software packages.

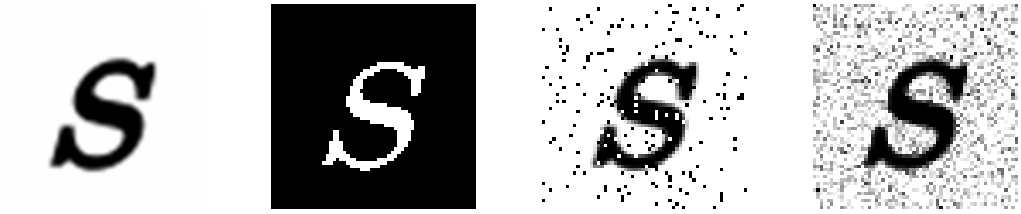
In recent years, multi-function-printers (MFPs) and high-speed scanners have been available, and customers demand various applications, such as OCR, over-scan, automatic document classification [5], skew correction [6], text to speech (TTS) [7] and region of interest (ROI) scan.

Regarding the performance of OCR, not only recognition accuracy but also processing speed is very important. For example, the automatic document feeder (ADF) for the Samsung Smart Multi-Xpress 7 series MFP can scan 240 pages a minute. The OCR processing speed should be fast enough to follow the scanning speed.

Character recognition or classification exploits pattern recognition and machine learning technologies. According to Ko *et al.* [8], they compared OCR methods for recognition accuracy and processing time by convolutional neural networks [11-13] and Tesseract [9, 10]. In addition, they proved that the convolutional neural network is suitable for character recognition.

Actually, a learning-based system is primarily dependent on the size and quality of the dataset. The Mixed National Institute of Standards and Technology (MNIST) [1] dataset achieved the highest performance in a study by Wan *et al.* [14], and the size of MNIST characters was 32 x 32 pixels. Thus, we have built a new dataset of characters that comprises normal, italic, and bold types with various fonts, such as Arial, Cambria, Consolas, Gothic, Times New Romans, and etc. In addition, the character size in the newly built dataset is 60 x 60 pixels. We named it the SP dataset.

The goal of this work is to analyze the performance of state-of-the-art convolutional neural network (deep-



(a) (b) (c) (d)

Fig. 1. An example of characters used in the experiments (a) noiseless character, (b) black-on-white text without noise, (c) character with salt-and-pepper noise at 10%; (d) character with Gaussian noise (ε is 0.00001).

Table 1. The list of SP datasets.

Dataset	Image Components	The Number of Characters	Description
SP-OCR	Digits Upper-and lower-case alphabets	18,600 characters	White-on-black text characters without noise
SP_R-OCR	Same as SP-OCR	Same as SP-OCR	Black-on-white text characters without noise
SP_N-OCR	Same as SP-OCR	Same as SP-OCR	White-on-black-text character with salt-and-pepper noise or Gaussian noise

learning) methods for single-character recognition, and we used the AlexNet, LeNet, and SPNet structures. Convolutional neural networks have recently attracted a lot of attention due to its superb recognition capability. To evaluate their performance, we had built a dataset of characters, because there are no publicly available datasets that contain lower-case characters. We experimented in the presence of salt-and-pepper noise or Gaussian noise, as seen in Fig. 1. Performance was measured in terms of recognition error (%).

The rest of this paper is organized as follows. In the next section, we describe the dataset for the experiments. In Section 3 is a brief overview of character recognition, and we delineate convolutional neural networks. In Section 4, we report the experimental results on the character recognition methods based on the AlexNet, LeNet, and SPNet structures. We also give an analysis of the performance comparison. Finally, in Section 5, we derive a conclusion and suggest future work.

2. Dataset

MNIST and Caffe-OCR [2, 3] are known by single-character dataset for character recognition [Fig. 2]. First, MNIST had only made up a handwritten digit dataset. Secondly, Caffe-OCR, which consists of upper-case letters and digits, was made up of black-on-white text.

These datasets are not appropriate for printed text recognition. So, we have built a new dataset, seen in Table 1. Our dataset contains the upper-case alphabet from A to Z, lower-case alphabet from a to z and digits from 0 to 9. The original image that included text was made using Microsoft Office Word and was segmented by our own segmentation tool. We generated 18,600 noiseless characters as SP-OCR dataset and characters with either salt-and-pepper noise or Gaussian noise as the SP_N-OCR dataset. Plus, we have built 18,600 black-on-white text characters.

(a) (b)

Fig. 2. Character sets for character recognition (a) MNIST, (b) Caffe-OCR.

3. Methods

In this section, we describe a convolutional neural network method under the AlexNet, LeNet and SPNet architectures.

Convolutional neural networks have emerged as an important area in artificial intelligence, machine learning and computer vision, due to rapid development in digital image processing with huge and high-quality datasets. The goal of a convolutional neural network method is to find a solution that best maps a set of correct output. Examples are handwritten text recognition [15], image classification [11, 12] and object detection [16] tasks.

Convolutional neural networks are perfectly validated by a huge dataset, such as CIFAR-10/100 [17], ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [18] and face recognition [19].

In the next sub-section, we will explain the architecture of AlexNet, LeNet and SPNet in detail.

3.1 AlexNet

Caffe is a deep-learning framework developed by the Berkeley Vision and Learning Center (BVLC) and by community contributors, and is openly available. AlexNet is only a small part of the deep-learning structure. It also

offers the Caffe framework.

AlexNet was constructed nine layers with five convolutional layers. The weights of initialization is Gaussian distribution, and the standard deviation used 0.01.

For AlexNet evaluation, we modified an AlexNet structure, as seen in Fig. 4(a). A description of the AlexNet architecture follows:

Description of the AlexNet architecture:
a. Input image size of 60x60
b. A convolutional layer by 96 maps with 11x11 kernels
c. A max pooling layer non-overlapping regions of size 3x3
d. A convolutional layer by 256 maps with 5x5 kernels
e. A max pooling layer non-overlapping regions of size 3x3
f. A convolutional layer by 384 maps with 3x3 kernels
g. A convolutional layer by 384 maps with 3x3 kernels
h. A convolutional layer by 256 maps with 3x3 kernels
i. A max pooling layer non-overlapping regions of size 3x3
j. A fully connected layer with 256 units
k. An output layer with 62 neurons (softmax activation function)

For performance evaluation, we used five-fold cross-validation. The characters were split into five groups, and we have trained and tested each character five times. Each time, four of the five groups were used as training data and another group was used as test data. The AlexNet training result is shown in Fig. 3.

It seems that the AlexNet structure cannot be optimized for a character recognition system. Actually, the AlexNet structure mostly uses a lot of image classification. The Caffe framework additionally included the AlexNet structure for image classification as CIFAR 10/100.

3.2 LeNet

LeNet is regarded as one of the most famous convolutional neural networks. The convolutional neural network method was improved by LeCun *et al.* [20]. In their work, a new framework for an MNIST digit recognition method was proposed with LeNet-5, which comprised seven hidden layers. To evaluate its performance, we revised the LeNet architecture as seen in Fig. 4(b). Input size, fully connected layer, and output layer were modified to use the SP_OCR dataset. A description of the LeNet architecture follows:

Description of the LeNet architecture:
a. Input image size of 60x60
b. A convolution layer by 20 maps with 5x5 kernels
c. A max pooling layer non-overlapping regions of size 2x2
d. A convolution layer by 50 maps with 5x5 kernels
e. A max pooling layer non-overlapping regions of size 2x2
f. A fully connected layer with 500 units
g. An output layer with 62 neurons (softmax activation function)

3.3 SPNet

SPNet is an approach designed by ourselves. Our goal is obviously to get high recognition accuracy using convolutional neural networks. Therefore, we efficiently refer to the LeNet structure that conducted hand-written character recognition.

Our structure was designed as a seven-layer deep structure with the same LeNet kernels [Fig. 4(c)], a slight variant of the LeNet structure. However, we used 3 x 3 max pooling layers and add convolution and pooling layers.

For experiments, we have given as the initial learning rate, 0.01; momentum, 0.9; decay, 0.0005; and feedback on work for every 100 epochs (1 epoch is 62 batches). In addition, we used summation of the squares of output for the ReLU (Rectified Linear Unit) after convolutional layers. Particularly, the ReLU is a very important part of the experiments. Both AlexNet and SPNet require ReLU handling after every convolutional layer and before the fully connection step, but LeNet does not, except for the fully connection previous step. Actually, the ReLU has the advantage in gradient efficiency of propagation, computing efficiency, scale-invariant and sparse activation. Plus, we clearly have to make mention of local-response-normalization (LRN). In fact, the current state-of-the-art deep-learning methods exclude the LRN step, and it is just a trend in convolutional neural networks. However, the SPNet is resolutely utilized to normalize every next step of the pooling layer, and brings a fine result for error rate.

The result of the SPNet training is shown in Fig. 3, and a description of the SPNet follows:

Description of the SPNet architecture:
a. Input image size of 60x60
b. A convolution layer by 20 maps with 5x5 kernels
c. A max pooling layer overlapping regions of size 3x3
d. A convolution layer by 50 maps with 5x5 kernels
e. A max pooling layer overlapping regions of size 3x3
f. A convolution layer by 50 maps with 5x5 kernels
g. A max pooling layer overlapping regions of size 3x3
h. A fully connected layer with 200 units
i. An output layer with 62 neurons (softmax activation function)

4. Experimental Results

Our experiments were conducted with the Caffe (deep-learning) framework, and our testing environment desktop included Microsoft Windows 7 (64-bit), and an Intel Core i5-2320 with 8GB and solid state drive (SSD).

4.1 Evaluating the SP-OCR Dataset

The SP-OCR dataset is a set of single-character images of 60 x 60 pixels. It contains 62 labels, such as the upper-case alphabet A to Z, lower-case alphabet a to z, and digits 0 to 9, the number of each training samples are 14,880, and the number of each test samples are 3,720.

The first experiments used an SP-OCR dataset that has noiseless characters. We reported the recognition error (%) at rank-1, rank-2 and rank-3. For the test, we used a trained-net that had the lowest recognition error between 6,000 and 7,000 epochs from the training data. The results with SP-OCR are shown in Table 2.

The misrecognition samples keep making the same errors over and over again, as seen in Fig. 5. The SPNet especially showed recognition errors with similar characters. An upper-case letter I was recognized as a

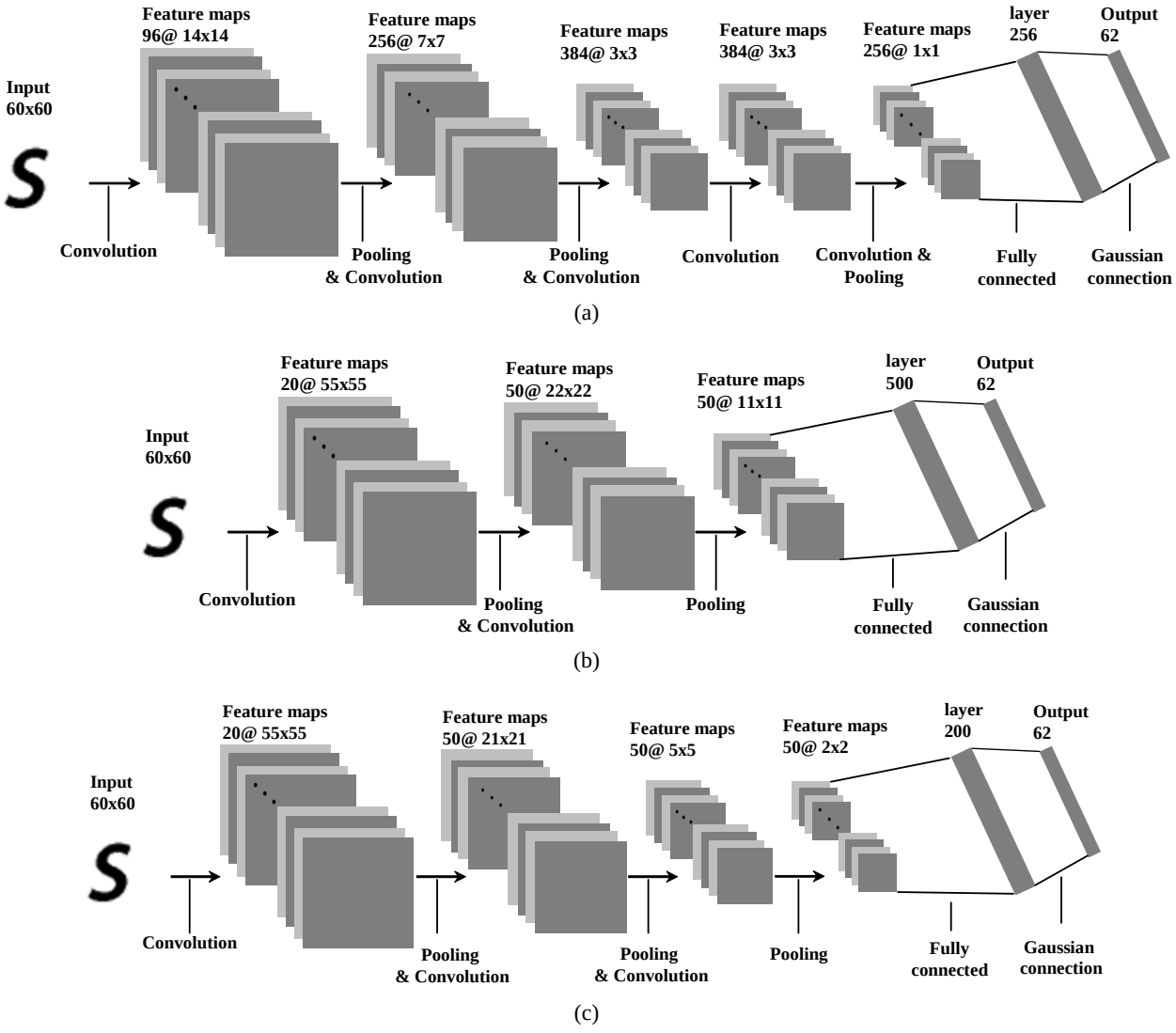


Fig. 4. Architecture of (a) CaffetNet, (b) LeNet, (c) SPNet.

Table 2. SP-OCR error rate (%).

Nets	Rank-1	Rank-2	Rank-3
AlexNet	0.0882	0.0305	0.0160
LeNet	0.0636	0.0559	0.0547
SPNet	**0.0420**	**0.0162**	**0.0113**

lower-case letter l and the natural number 1 was recognized as a lower-case letter l, the reverse also appeared. In fact, in this error, it is a difficult task to distinguish the lower-case letter l and upper-case letter I.

4.2 Evaluating the SP_R-OCR Dataset

The second experiment used the SP_R-OCR dataset, which consists of black-on-white text. We also report recognition error (%) as rank-1, rank-2, and rank-3 in Table 3.

The tendency for recognition error with each character was the same as the experiment for the SP-OCR dataset [Fig. 6].

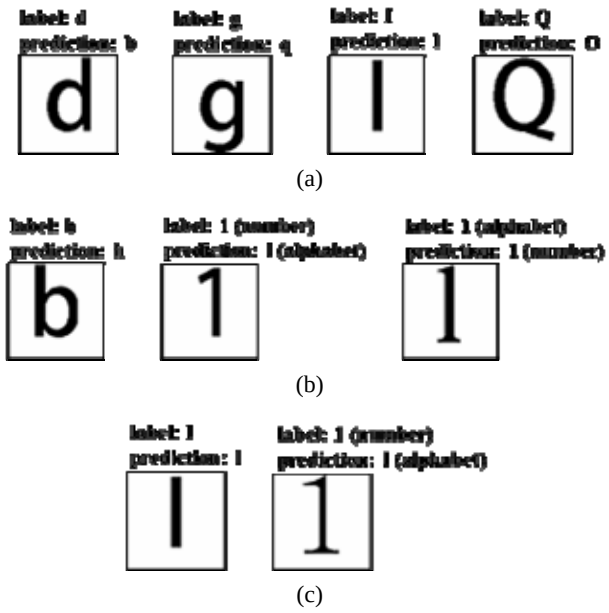


Fig. 5. Examples of misrecognition by SP-OCR (a) AlexNet, (b) LeNet, (c) SPNet.

Table 3. SP_R-OCR error rate (%).

Nets	Rank-1	Rank-2	Rank-3
AlexNet	0.0885	0.0305	0.0166
LeNet	0.1261	0.1147	0.1128
SPNet	**0.0528**	**0.0241**	**0.0165**

(a)

(b)

(c)

Fig. 6. Examples of misrecognition by SP_R-OCR (a) AlexNet, (b) LeNet, (c) SPNet.

Table 4. SP_N-OCR error rate (%).

Nets	Rank-1	Rank-2	Rank-3
AlexNet	0.2274	0.1214	0.0823
LeNet	0.1500	0.1353	0.1320
SPNet	**0.2784**	**0.1948**	**0.1624**

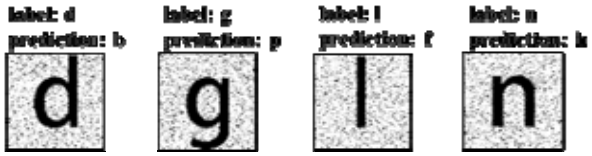
From the results with the SP_R-OCR dataset, we can confirm that white-on-black text is better for recognition accuracy, but for LeNet, that is not the case. The SPNet also showed the best performance when the SP_N-OCR dataset was evaluated.

4.3 Evaluation of the SP_N-OCR Dataset

The final experiments used noisy characters in the SP_N-OCR dataset. For testing, we used trained_net data from SP-OCR, which means no training process.

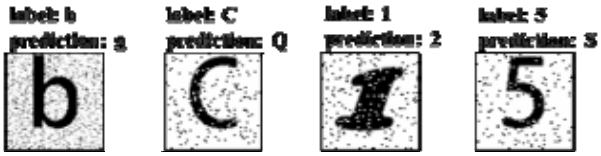
The results for SP_N-OCR are shown in Table 4, and examples of misrecognized characters are shown in Fig. 7. The tendency for recognition error of each character is not seen, but g was misrecognized as p, B was misrecognized as 7 and b was misrecognized as g. Every net showed that it has one great weakness: noise. Obviously, the SPNet (explicitly, every net) has a fatal weakness, and will be unavailable under noise.

For noisy character recognition, the character has to be processed noise removal filter.



(a)

(b)



(c)

Fig. 7. Examples of misrecognition by SP_N-OCR (a) AlexNet, (b) LeNet, (c) SPNet.

Table 5. Recognition times: average value for all-experiments.

Nets	Recognition Time (ms) for one character
AlexNet	4.31
LeNet	**2.15**
SPNet	2.58

4.4 Recognition Times

To evaluate OCR performance, not only recognition accuracy but also recognition time is very important components. For measuring time, we have to record the recognition time for all the experiments, and Table 5 gives the results of our testing environments. For processing time capability, the LeNet structure (which consists of seven hidden layers) showed the best performance.

5. Conclusion

Our main approach is to analyze the performance of state-of-the-art convolutional neural network methods for character recognition. The architectures for AlexNet, LeNet and SPNet were used to accomplish the tasks, and we additionally have built a dataset for single-character recognition and provided this dataset. The results of recognition error are utilized to compare the performance of AlexNet, LeNet and SPNet as shown in Fig. 8.

As a result, convolutional neural networks have sufficiently proven that most characters can be perfectly recognized, and our approach (the SPNet structure) surpasses the state-of-the-art convolutional neural

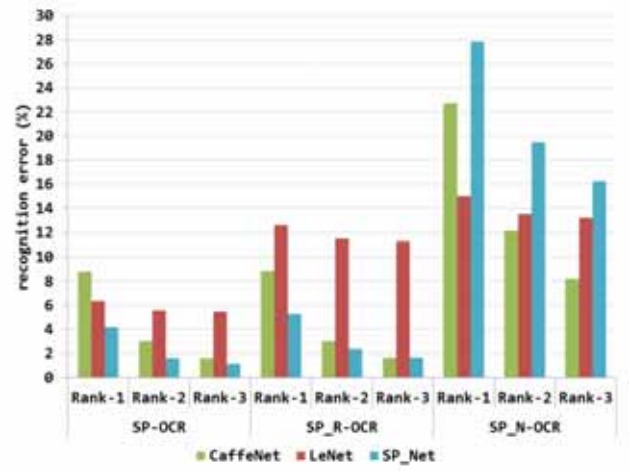


Fig. 8. Summary by experimental result.

(a) (b)

Fig. 9. The size of lower-case letters (a) and upper-case letters (b) are the same.

networks for character recognition. However, especially for recognition accuracy, several cases, (such as upper-case letter I, lower-case letter l, upper and lower case letter O, and the number 1), will need to be improved in the future. For this, a digits group and upper-case and lower-case alphabet groups will be classified first, and we will then proceed to recognize each group.

In addition, our study can be extended to a normalized character dataset (the same as resizing characters in a fixed image), since it possibly could have more information or features in an image, as shown in Fig. 9. Therefore, we will be ready to normalize the dataset and again evaluate performance under AlexNet, LeNet and SPNet.

Moreover, we will utilize the method for character recognition and challenge sentence classification [21-23].

References

[1] Y. LeCun, C. Cortes, and C. J. C. Burges. The MNIST Database of Handwritten Digits, Article (CrossRef Link)

[2] Caffe-OCR. OCR with Caffe Deep Learning Framework, Article (CrossRef Link)

[3] BLVC (Berkeley Vision and Learning Center). Caffe Deep Learning Framework, Article (CrossRef Link)

[4] R. Mither, S. Indalker, and N. Divekar. Optical Character Recognition, *International Journal of Recent Technology & Engineering*, IJRTE, 2013. Article (CrossRef Link)

[5] 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. Article (CrossRef Link)

[6] Y. Cao, S. Wang, and H. Li. Skew detection and correction in document images based on straight-line fitting, *Pattern Recognition Letters*, ELSEVIER, 2003. Article (CrossRef Link)

[7] A. Kain, and M. W. Macon. Spectral voice conversion for text-to-speech synthesis, *Proceedings of the 1998 IEEE International Conference on*, IEEE, 1998. Article (CrossRef Link)

[8] D. G. Ko, S. H. Song, K. M. Kang, S. W. Han, and J. H. Yi. Optical Character Recognition Performance Comparison of Convolutional Neural Networks and Tesseract, *The 31st International Technical Conference on Circuits/Systems, Computers and Communications Technical Program*, ITC/CSCC: pp. 871-874, 2016. Article (CrossRef Link)

[9] R. Smith. Tesseract OCR Engine, *Google Inc.*, OSCON, 2007. Article (CrossRef Link)

[10] R. Smith. An Overview of the Tesseract OCR Engine, *International Conference on Document Analysis and Recognition*, IEEE, 2007. Article (CrossRef Link)

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks, *in Neural Information Processing System*, NIPS, 2012. Article (CrossRef Link)

[12] D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column Deep Neural Networks for Image Classification, *in Proceedings of CVPR 2012*, IEEE, 2012. Article (CrossRef Link)

[13] Y. LeCun, L. Bottou, Y. Bengoi, and P. Haffner. Gradient based learning applied to document recognition, *in Proceedings of the IEEE*, IEEE, 1998. Article (CrossRef Link)

[14] L. Wan, M. Zeiler, S. Zhang, Y. LeCun, and R. Fergus. Regularrization of Neural Networks using DropConnect, *in Proceedings of the 30th International Conference on Machine Learning*, ICML, 2013. Article (CrossRef Link)

[15] D. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber. Deep, big, simple neural nets for hand written digit recognition, *Neural Computation*, MIT Press Journals, 2010. Article (CrossRef Link)

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchites for accurate object detection and semantic segmentation, *in Proceedings of CVPR 2014*, IEEE, 2014. Article (CrossRef Link)

[17] A. Krizhevsky. Convolutional Deep Belief Networks on CIFAR-10, *Unpublished manuscripts*, 2010. Article (CrossRef Link)

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks, *in Neural Information Processing System*, NIPS, 2012. Article (CrossRef Link)

[19] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep Face Recognition, *in the British Machine Vision Conference*, BMVC, 2015. Article (CrossRef Link)

[20] Y. LeCun, L. D. Jackel, L. Bottou, C. Cortes, J. S. Denker, H. Denker, I. Guyon, U. A. Muller, E. Sackinger, P. Simard, and V. Vapnik. Learning

Algorithms for Classification: A Comparison on Handwritten Digit Recognition, *Neural Networks: The Statistical Mechanics Perspective*, World Scientific: 261-276, 1995. Article (CrossRef Link)

[21] Y. Kim. Convolutional neural networks for sentence classification. *in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, EMNLP: pp. 1746-1751, 2014. Article (CrossRef Link)

[22] R. Johnson and T. Zhang. Effective use of word order for text categorization with convolutional neural networks. CoPR, 2014. Article (CrossRef Link)

[23] X. Zhang, J. Zhao, and Y. LeCun. Character-level Convolutional Networks for Text Classification, *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS, 2015. Article (CrossRef Link)

Dae-Gun Ko received his BSc in Electronic Engineering and Computer Engineering from Yeongnam University, South Korea, in 2009, and hold a Samsung Electronics Software Membership from 2006 to 2009. He received the MSc from the Department of Digital Media and Communications Engineering at Sungkyunkwan University, South Korea, in 2016. He is currently a researcher at Samsung Electronics Co.,Ltd., South Korea. His research interests include image processing, pattern recognition, deep-learning, and computer vision systems for robots.

Su-Han Song received his BSc with honors in Electric Electronic & Computer Engineering from Hanyang University, South Korea, in 2006, where he received his M.Sc. with honors in Electronics, Communications & Computer Engineering in 2008. He is currently working as a researcher at Samsung Electronics Co. Ltd., Suwon, South Korea. His current interest focuses on improving automatic image defect detection in manufacturing processes using machine vision.

Ki-Min Kang received his PhD in Electrical Engineering from Inha University, South Korea, in 2001. He joined Samsung Electronics Co. Ltd., in 2001, and researched and developed the algorithms and the pipelines related to image enhancement and quantitative quality diagnosis with vision system. Now, he develops documents workflow solutions embedded in the copier machines based on scene analysis and optical character recognition.

Seong-Wook Han received the BSc in Electronics Engineering from Yonsei University, Seoul, Korea, in 2000, and the MSc in Electrical and Electronics Engineering from Yonsei University, Seoul, Korea, in 2002. From 2002 to 2004, he was a research engineer at on Timetek Inc., Seoul, where he worked on video compression, video transmission, and pre/post processing for digital broadcasting systems. In 2009, he received his PhD in electrical engineering from Purdue University, West Lafayette, IN. Since January 2009, he has been with Samsung Electronics Co. Ltd., Suwon, Korea, developing algorithms for electronic imaging systems. His research interests include electronic imaging systems, color processing, video coding, image/video analysis and image enhancement.