

Low-Quality Banknote Serial Number Recognition Based on Deep Neural Network

Unsoo Jang*, Kun Ha Suh*, and Eui Chul Lee**

Abstract

Recognition of banknote serial number is one of the important functions for intelligent banknote counter implementation and can be used for various purposes. However, the previous character recognition method is limited to use due to the font type of the banknote serial number, the variation problem by the solid status, and the recognition speed issue. In this paper, we propose an aspect ratio based character region segmentation and a convolutional neural network (CNN) based banknote serial number recognition method. In order to detect the character region, the character area is determined based on the aspect ratio of each character in the serial number candidate area after the banknote area detection and de-skewing process is performed. Then, we designed and compared four types of CNN models and determined the best model for serial number recognition. Experimental results showed that the recognition accuracy of each character was 99.85%. In addition, it was confirmed that the recognition performance is improved as a result of performing data augmentation. The banknote used in the experiment is Indian rupee, which is badly soiled and the font of characters is unusual, therefore it can be regarded to have good performance. Recognition speed was also enough to run in real time on a device that counts 800 banknotes per minute.

Keywords

Banknote Recognition, Convolutional Neural Network, Machine Learning, Optical Character Recognition, Serial Number Recognition

1. Introduction

In recent years, a steady decline has been witnessed in the use of banknotes owing to the development of electronic means for conducting financial dealing. However, the transactional importance of banknotes has not decreased, and dealing directly involving them are expected to continue. Banknotes will continue to remain important provided that their monetary value is maintained, which depends on the achievement of key requirements such as transactional safety, mass and automation processing.

To meet such requirements, machines suitable for banknote trading (machines that automatically recognize, calculate, and store banknote images) have been developed, such as banknote counters. It is typically used for bank clerk's jobs and is also installed for automation machines (vending machines, cash machines, etc.). The machine carries out the complex functions such as recognizing banknote, determining whether they are counterfeit, and batching banknote in large quantities, to achieve the many

※ This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Manuscript January 16, 2018; first revision April 9, 2018; accepted November 27, 2018.

Corresponding Author: Eui Chul Lee (ecllee@smu.ac.kr)

* Dept. of Computer Science, Sangmyung University, Seoul, Korea ({dnstn333, tjrjsgk}@naver.com)

** Dept. of Human Centered Artificial Intelligence, Sangmyung University, Seoul, Korea (ecllee@smu.ac.kr)

demands of banknote trading.

Banknote recognition through serial numbers enables cash flow tracking, illegal copying of banknotes, and forgery of parts of banknotes. In order to perform these various functions, serial number recognition may be an important function of a banknote counter.

Due to the nature of operating in an embedded system environment, traditional serial number recognition technology uses the classic optical character recognition (OCR) algorithms such as those used for license plates or document recognition [1,2]. Classical OCR is generally applied to banknotes without relatively background, a representative example of which is the banknote used in the United States and China, as shown in Fig. 1(a) and 1(b). However, because of low recognition rates the classic OCR cannot be used on paper notes with a lot of noise, as shown in Fig. 1(c).

Therefore, we propose a method for extracting the region of interest (ROI) of a banknote on the basis of its aspect ratio and then a method for extracting single character images using a recognition method based on a convolutional neural network (CNN). Our experimental results show that the average recognition performance rate of the proposed methods for a single character is 99.85%.

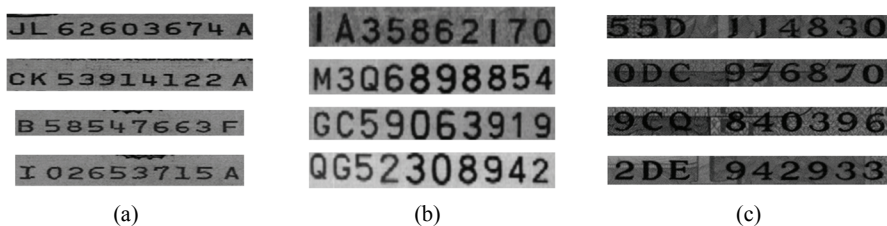


Fig. 1. Serial number image by country: (a) US dollar, (b) Chinese yuan, and (c) Indian rupee.

2. Related Work

During the last few years, research has been conducted on character recognition in various environments and applications. In this section, we categorize the types of character recognition systems that have emerged as a result of various research studies and describe each type.

Character recognition systems can be classified based on the manner in which the data, whether time series data or the font limit, are acquired. Fig. 2 visualizes the classification performed by a character recognition system. Based on the input type, character recognition systems can be categorized as “handwriting recognition” and “machine-printed character recognition” systems. Handwriting recognition is very difficult because every user moves the pen differently to produce the same character. Handwriting recognition systems can be classified into two subcategories: online and offline.

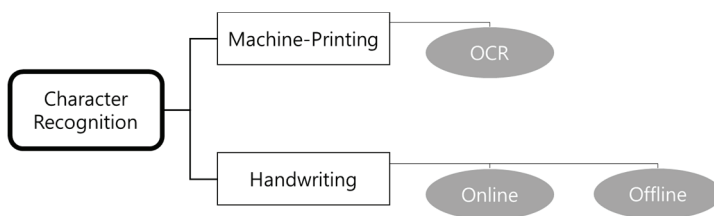


Fig. 2. Classification of a character recognition system.

Online recognition systems are operated in real time while the user is writing. They are less complex than offline systems because they can capture time-based information, i.e., the speed and tempo, the direction of strokes, etc. In addition, the size of the characters is sufficiently small to obviate the need for thinning. Because they use static, e.g., bitmap, data, offline recognition systems cannot easily achieve a high performance. As a result, online systems may be used more than offline systems. Online systems are easy to develop, are highly accurate and can be integrated into the input of tablets and personal digital assistants [3].

In contrast, “machine-printed character recognition” is a relatively simple problem because the size of the characters is uniform and their position can be predicted [4]. OCR, a machine-printed character recognition method, plays a major role in digitizing analog documents. However, among the analog documents, banknotes present the following constraints as compared to other documents. First, banknotes are used frequently and are thus subject to various types of physical damage. Second, when banknote data are required it is frequently necessary to process large amounts of data at high speed. Because of these constraints, the algorithms that can be applied to banknote serial number recognition are limited. In the following paragraph, the OCR method for banknote serial number recognition is described.

Most banknote serial number recognition algorithms follow the same flow as the traditional OCR method. First is the preprocessing step for the banknote image. In this step, the image normalization process and segmentation by character unit are performed to improve the visibility of the character region. Next, a feature vector is detected in each character area. Finally, the pattern classification technique for the features is used to recognize the character. That is, using a hand crafted feature defined by the developer.

Image processing techniques used for preprocessing include mean filtering for eliminating noisy pixel values [1,5], brightness/contrast/gamma adjustment [6], size normalization by bilinear interpolation [7], and grayscale normalization or binarization [8]. The algorithms used in the preprocessing step are simple and frequently used in combination. A combined algorithm is used to eliminate noise and obtain only the character area or to segment each character. In other words, the image processing method facilitates effective character feature extraction. For example, Penz et al. [5] proposed a serial number recognition method for Austrian banknotes, in which the features are extracted from a binarized image. The preprocessing algorithm for the extraction procedure involves interpolation for increasing the resolution, mean filtering, and finally binarization to obtain a high-quality binarized image [5]. Similarly, we propose in this paper a method that also involves multiple preprocessing techniques, such as binarization and size/skew normalization, to ultimately obtain a grayscale single character image with no margins.

We now discuss existing feature extraction and character recognition methods.

In general, character recognition methods can be classified into three types. Template matching, statistical, and structural analysis methods are the three types. The first is template matching. To match a template, a template for each character must first be registered; Then compare the pattern of the character you want to recognize with the pattern of the pre-registered template to determine the most similar character [9]. Template matching has the advantage of being quick and easy to implement because the algorithm is simple. However, not only does the method require that a template pattern be maintained for each character and format, but it must also be sensitive to noise Secondly, statistical methods [1,10] recognize characters based on the vector space distribution. The vector space distribution consists of features defined to distinguish each character. In the statistical method, selection of appropriate feature vectors, dimension determination, and classifier selection are important in terms of recognition accuracy and processing speed. Therefore, it is important to define the appropriate function for the data and design the classifier in the vector space of the defined function. In particular, proper consideration of lighting

conditions and noise factors for banknote images is also essential. Finally, there is structural analysis method that analyzes the character to be recognized to find out the construction rules and analyzes the relationship between the components of the character [2,11]. The problem with the structured method is that it requires basic analysis of the text, as well as information on all the rules that can be used to recognize the text.

In this paper, we propose a character recognition method using the statistical method described above. We propose a method to extract characters that are suitable for the environment and to recognize characters by using the extracted features as classification criteria. By performing feature selection and classifier design using neural network technology, we overcome the shortcomings of traditional methods. Neural network-based approaches have been actively adopted in various fields because of their good performance [12-14]. In our task, more specifically, a CNN technique is applied to feature selection to automatically select the appropriate features for data recognition. Further, in the design of the classifier for character recognition a neural network technique involving fully connected layers is applied. In the next section, the proposed method is described in detail.

3. Proposed Method

The proposed method includes image segmentation and character recognition. Image segmentation is performed based on the aspect ratio of the banknote, and CNN is used for feature extraction and character recognition. The obtained banknote image is converted into a rectangle, in which a simple image processing method is used. And the image of each character is extracted from the serial number position (ROI) peculiar to a banknote. In addition, the characters in each character image are recognized using trained CNN. Finally, character recognition results are obtained for each character. The flowchart of the proposed method is shown in Fig. 3.

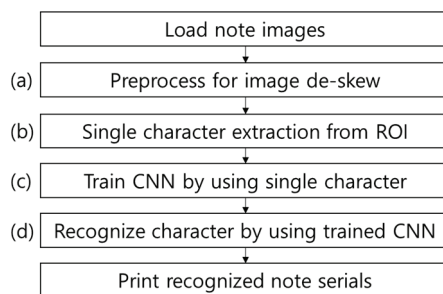


Fig. 3. Flowchart of the proposed method.

Steps (a), (b), and (c) of the flowchart are now described in detail. First, we explain the preprocessing step, i.e., step (a). The images obtained from the banknote counter are not uniform in shape owing to the flapping of the banknotes and are not precisely rectangular. Therefore, to obtain the serial number region (the ROI), we apply a de-skewing algorithm to the banknote images. The de-skewing algorithm proceeds in the order of binarization, edge detection, and affine transformation, as shown in Fig. 4. To de-skew an image, first the input image is binarized using the p -tile method based on the image [15]. p -tile is a method for determining the threshold value such that the ratio of objects after binarization converges to a predetermined p when the ratio of the size of the object in the image is known. Then, four contacts

(corners of the banknote) of the detected edge are used to calculate the affine transform matrix. Descale is performed by using the computed affine matrix. Distortion removal is performed based on the inherent aspect ratio of the banknote. The ROI's position of the banknote is estimated based on the inherent aspect ratio of the note, using the backward mapping method to approach the ROI position to obtain a de-skewed ROI image.

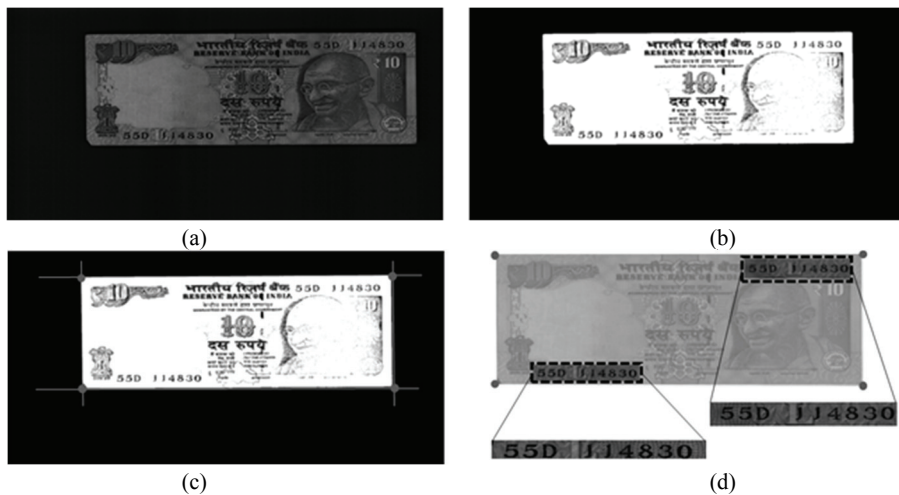


Fig. 4. Visualization of the de-skew algorithm: (a) input image, (b) binarized image, (c) find corner points, and (d) extract region of interest.

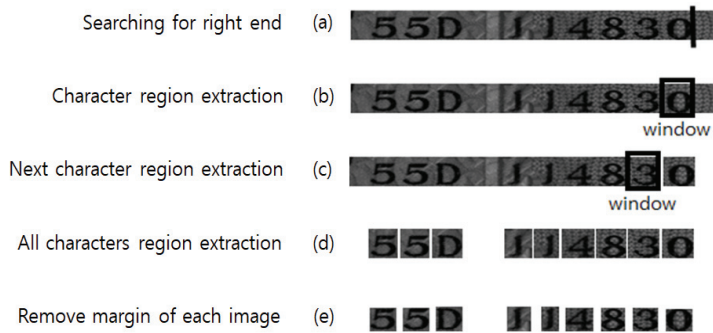


Fig. 5. Visualization of character segmentation algorithm.

In step (b), the serial number in the ROI is segmented. In the case of Indian rupee bills, the ROI includes a noisy background, rendering the application of existing segmentation methods difficult. Therefore, the position of the ROI is inferred using the aspect ratio of the banknotes and finely adjusted. The character segmentation algorithm is shown in Fig. 5 and described in detail as follows. First, the character is checked from the right end in order to split one character at the right end. Segmentation is performed by the inferred size based on the aspect ratio. A window of constant size is used to crop the image to include all of character. A window of constant size is positioned to ensure that there is empty space at the inferred position and that the empty space is at its maximum. Next, the empty spaces on each extracted character image are removed. All character images with empty space removed are the same size. This is because all character is printed on the same size, regardless of the banknote and character type.

Finally, step (c) is now explained. Because serial number ROIs include background patterns in shown as Fig. 4(d), existing simple algorithms cannot easily achieve high recognition rates. In contrast, by adopting complex algorithms, we can achieve satisfactory OCR performance. However, complex algorithms are not suitable for embedded systems that require bulk high-speed processing. Therefore, in the proposed method, we obtain high performance with simple calculation by using neural network technique. The neural network design for character recognition is shown in Fig. 6. An analysis of various neural networks is provided in Section 4.

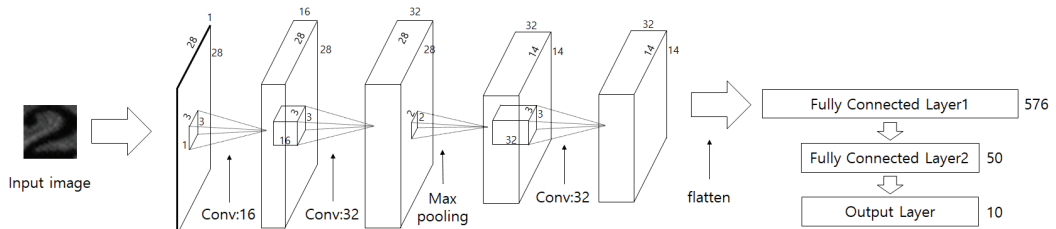


Fig. 6. Neural network model obtained for character recognition.

The training step of the neural network is as follows. A CNN method is utilized to learn the feature model using convolutional layers, and the classification model is learned using fully connected layers [16]. To prevent overfitting in training, we added a dropout layer in the fully connected layers of the neural network. The dropout probability is 50%. The neural network is trained using both augmented data and original data and thus it yields two types of results. In this study, data augmentation, which is discussed in detail in section 4, was used to reduce the incidence of error cases. The data augmentation was performed by random parallel shifting of the single character image data up to 0.05% vertically and 0.2% horizontally. To learn the model, we divided the entire character image data into three sets (training, validation, and testing). The three sets were created by dividing randomly the entire image data by a ratio of 70%, 15%, and 15% for training, validation, and testing, respectively. This data division policy was applied equally to all models. The training step was run for 50 epochs with a batch size of 256. During training, if the loss value of the validation set was not improved in the predetermined epoch, the training was stopped. This stopping is called early stopping [17].

Fig. 6 shows the internal structure of the neural network used for recognition. The training proceeds in the order of two convolutional layers, one pooling layer, and one convolutional layer. In the case shown in Fig. 6, the numbers of kernels of the convolutional layers are 16, 32, and 32, respectively. In the convolutional layers, a 3×3 mask is applied at a 1-pixel stride. Further, we use max pooling for the pooling layer and apply a 2×2 mask at a 1-pixel stride. In addition, we use ReLU as the activation function of the nodes constituting the neural network.

For the CNN training and testing in this study, the hardware used was an Intel core i7 and NVIDIA GeForce GTX 650 and the software was Microsoft Windows 10 (64-bit), Python, and Keras.

4. Experiments

4.1 Experimental Data

For experiment, images of real banknotes were collected by scan image capture, and nine types of

banknotes were collected. The number of notes, actual note size, and serial number area size of each type are summarized in Table 1. Here, “new notes” refers to those issued in 2005. The “serial number image sizes” means the size of the serial number area as shown in Fig. 4(d), which contains 9 characters. In the process of geometric transformation for the de-skewing of the serial number region, backward mapping method including bilinear interpolation strategy is used to prevent hole, overlapping and aliasing. Because all serial number fields were identified with the same ROI location and size, regardless of note type, the serial number regions were normalized to the same size.

Table 2 lists the number of single characters extracted from the ROI through character segmentation. The size of each single character image is 28×28 pixels. In the character segmentation step, the same segmentation size was used for all types of characters and notes, because the size of all the characters (except for 1) on the banknotes is nearly the same.

Table 1. Information on the banknote images used in the experiment

Type of notes	Number of notes	Real note sizes (mm×mm)	Serial number image sizes (pixel×pixel)
10	395	117×73	(about) 304×30
20	314	137×73	(about) 304×30
50	342	147×73	(about) 304×30
100	222	157×73	(about) 304×30
(new note) 100	486	157×73	(about) 304×30
500	162	167×73	(about) 304×30
(new note) 500	228	167×73	(about) 304×30
1000	191	177×73	(about) 304×30
(new note) 1000	360	177×73	(about) 304×30

Table 2. Number of images per character extracted from banknote images

	Character									
	0	1	2	3	4	5	6	7	8	9
Number of images	957	580	687	780	572	1,784	999	1,049	613	575

4.2 Neural Network Design

To establish a suitable neural network design for the recognition of the serial numbers on banknotes, we designed four types of neural network, deep/shallow in terms of the number of layers and heavy/light in terms of the number of kernels, and compared their performance (see Fig. 7). The first model (deep/heavy) has a large number of kernels in each layer and a large number of layers. The second model (deep/light) has a small number of kernels for each layer but a large number of layers. The third model (shallow/heavy) has a large number of kernels in a layer, but a small number of layers. Finally, the fourth model (shallow/light) has a small number of kernels in each layer, and a small number of layers. In Section 4.4, the performance of each neural network is described in detail.

4.3 Data Augmentation

When the performance of the four types of neural networks trained using the original data was analyzed, error cases (see Fig. 8) were frequently observed. The images in Fig. 8 show some cases where an error occurred in the character segmentation step. It was observed that the deviation degree and frequency are greater along the horizontal than along the vertical axis.

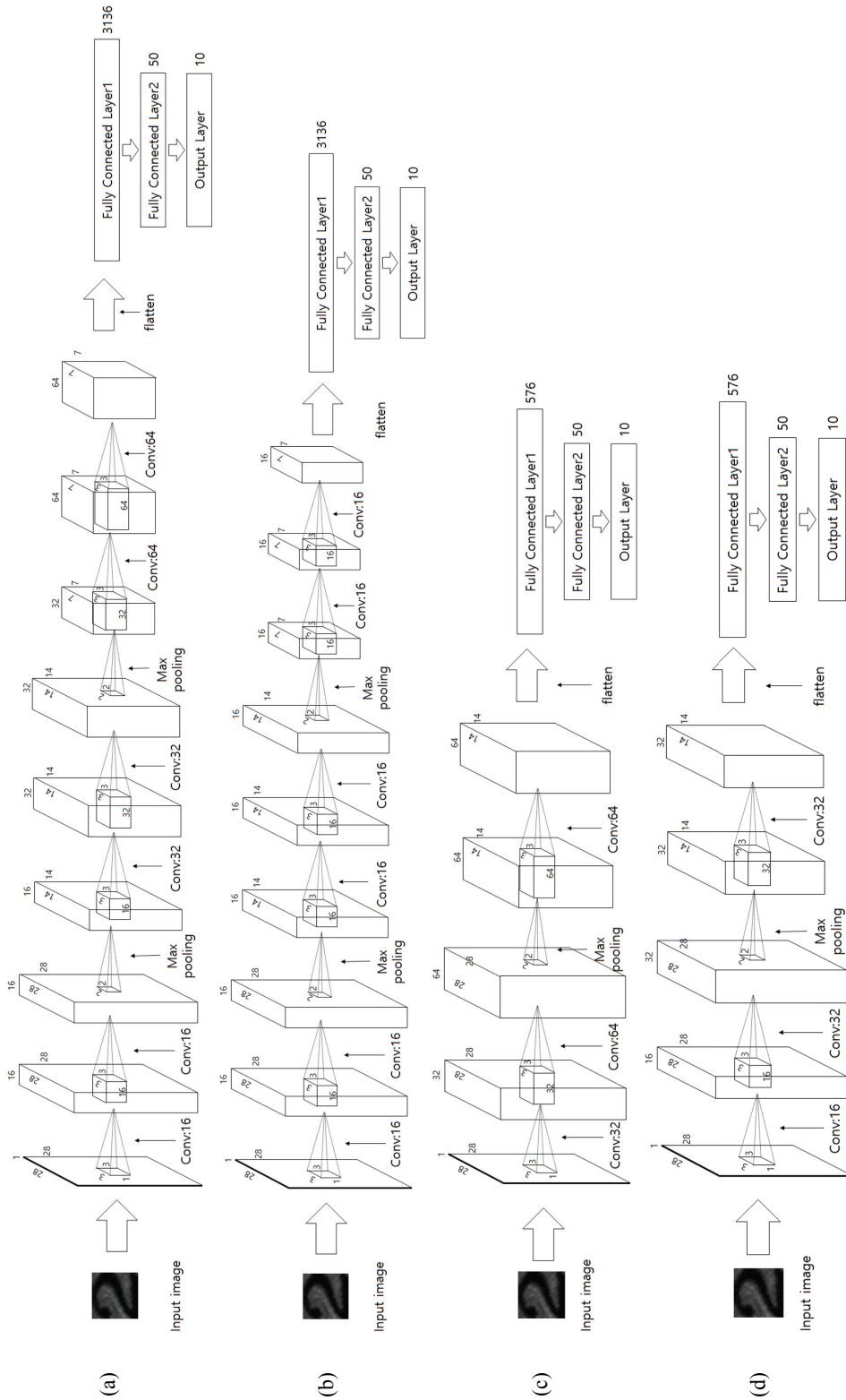


Fig. 7. Four types of neural network: (a) deep/heavy, (b) deep/light, (c) shallow/heavy, and (d) shallow/light.

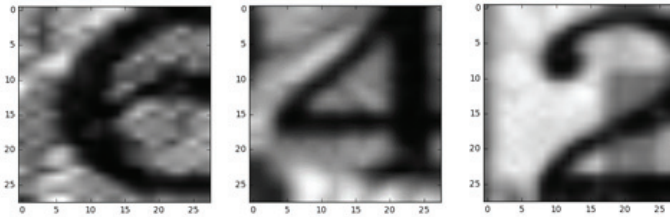


Fig. 8. Example images of error cases.

Therefore, to establish a robust feature extraction model and classification model for a single character image, we randomly generated 0.2% data in the areas to the left and right hand side of the single character image and 0.05% data in the areas above and below. At the time of data augmentation, the area where no image existed was filled with a constant value, as shown in Fig. 9.

As a result of the data augmentation, 16,000 images were generated from the original 6,000 images. The performance and analysis of the models in which augmented data were applied are discussed in the next subsection.

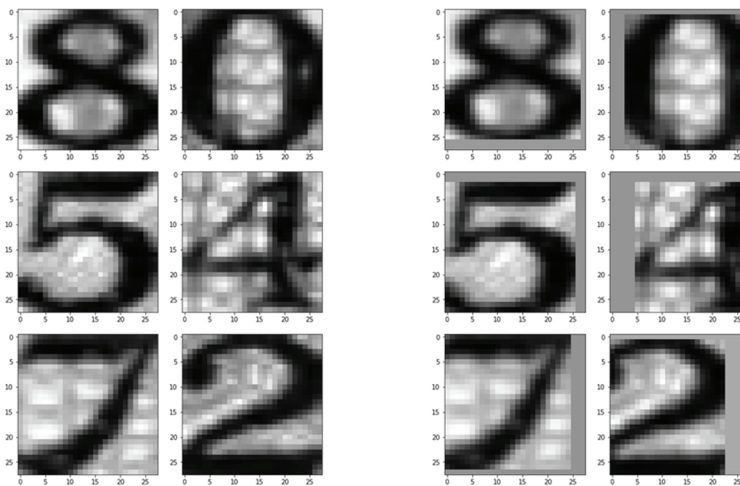


Fig. 9. Six example images before (left) and after (right) data augmentation using constant value.

4.4 Results and Analysis

The performances of the four neural networks proposed in Section 4.2 were compared and analyzed. The purpose was to select a neural network model suitable for single character recognition by means of the results analysis. Each neural network model yielded two results, those when augmented data were applied and those when the original data were applied. The experimental results that were used to select the appropriate neural network model are represented by an accuracy graph for the training set, a loss value graph for the validation set, and an accuracy table for the test set.

Fig. 10 shows the accuracy graph for the training set. In Fig. 10, a total line graph shows the progress of the entire learning trend and a magnified line graph shows the details of their congestion. The horizontal axis of the graph represents the epoch and the vertical axis the accuracy. For each epoch during which the model is being learned, the accuracy for the training set is calculated. The change in accuracy

according to the progress of training can be seen. The calculated accuracy in each epoch helps to estimate the extent to which each model has been learned and the manner in which each model has been changed. The model labeled “Deep&Light&DataAugm” is a representative model, in which the accuracy rises quickly and steadily. In contrast, the accuracy of the model labeled “Shallow&Light” slowly increases, repeatedly rising and falling, as compared with other models.

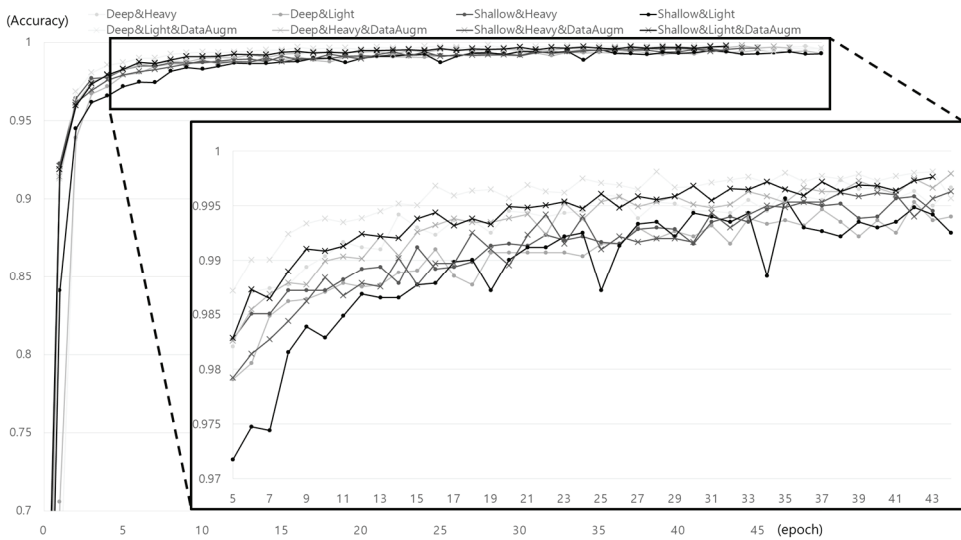


Fig. 10. Training set accuracy graphs of the four neural networks.

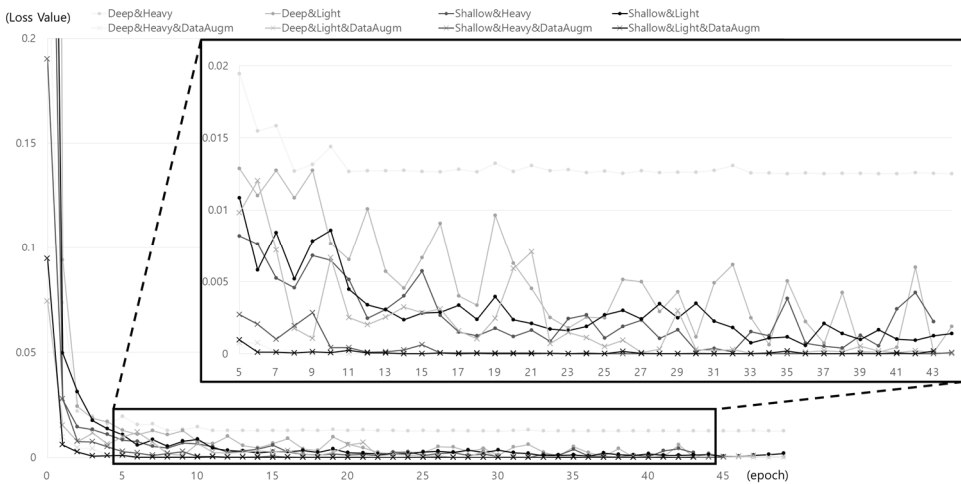


Fig. 11. Validation set loss value graphs of the four neural networks.

Fig. 11 shows the loss value graph for the validation set, presented in the same manner as the accuracy graph in Fig. 10. The horizontal axis represents the loss value and the vertical axis represents the epoch. The decrease in the loss value according to the progress in each epoch can be seen. For each epoch during which the model is being learned, the loss value for the validation set is calculated. The calculated loss value in each epoch helps predict the model’s stability and performance in real-world applications. The

model labeled “Deep&Light&DataAugm” is a representative model in which the loss value falls quickly and steadily. In contrast, the loss value of the model labeled “Deep&Light” slowly decreases, repeatedly rising and falling, as compared with other models. In addition, the model labeled “Deep&Heavy” shows instances where the value does not decrease below a certain value.

Finally, the accuracy of the test set before and after data augmentation was observed, as shown in Table 3. The accuracy of the test set is the result for the trained model after the final epoch, not for each epoch. In addition, the computational time of the four neural networks for training and testing was measured, the results of which are shown in Table 4. The computational time for training shown in Table 4 was calculated for the case where the augmented data set was used.

Table 3. Test set accuracy (%) of the four neural networks before and after data augmentation

Test set accuracy	Deep		Shallow	
	Heavy	Light	Heavy	Light
Original	99.92 (1/1290)	99.76 (3/1290)	99.69 (4/1290)	99.84 (2/1290)
Augmented	100.00 (0/1290)	100.00 (0/1290)	99.69 (4/1290)	99.92 (1/1290)

Table 4. Computational time (ms per banknote) of four neural networks for training and testing

Processing time	Deep		Shallow	
	Heavy	Light	Heavy	Light
Training	28.2	18.5	19.0	15.3
Testing	0.56	0.37	0.38	0.31

Now, we discuss the analysis of the results. First, the results illustrated in the two graphs described above (Figs. 9 and 10) show the effect of data augmentation. A comparison of the results before and after data augmentation shown in the two graphs reveals the following two characteristics. By applying data augmentation, first, the improvement in the performance per epoch was increased to reach the maximum performance in a short time, as shown in Table 4. Second, the deviation between epochs during the training was reduced. In addition, from the results of this comparison it can be inferred that the training process is more stable with than without data augmentation. In other words, the application of data augmentation causes the training to progress steadily and quickly. Further, the progress of the training leads us to assume that the training of the neural network is robust to errors in the images. In addition, it can be seen that the overfitting phenomenon is relatively reduced with the progress of the improved training. Thus, the application of data augmentation can be expected to improve the single character recognition performance.

Next, the performance of each (neural network) model was evaluated on the basis of the results shown in the two graphs (Figs. 9 and 10). For the reasons mentioned above, we focus on the model in which data augmentation was applied. To evaluate the performance of each model, we first compare the efficiency and stability shown in the loss value graph. For evaluating the performance of a model, three aspects can be considered: “Is there a relatively small variation in the progress of learning?” “Is the loss value of the model in general lower than that of the other models?” and finally “Do some frontal epochs show a dramatic improvement in performance?” According to these considerations, the evaluation of the models showed that the level of their performance follows the order shallow/light > deep/heavy > deep/light > shallow/heavy. The analysis of the evaluation procedure reveals that the models with relatively deep neural networks are stable but do not lead to a large difference in the results. In contrast,

we can see that the shallow/light model yields results that are similar to or more positive than those of the deep/heavy model.

The analysis of the test set accuracy of each model is described as follows. As shown in Table 3, the highest performance rate is 100%, and the performance is improved by data augmentation, except in the case of one model (shallow/heavy). Further, among the augmented data models, the highest character recognition performance was achieved using the deep model (Fig. 6(a) and 6(b)) that completed the learning. The shallow/light model showed a 99.92% performance rate, which is nearly equal to that of the deep model, whereas the shallow/heavy model showed a relatively low performance rate.

Because the size of the data used in the experiment was not sufficiently large, the test set accuracy may not be considered reliable. However, according to the two graphs the tendency of the test set accuracy is similar to that of the performance evaluation, and it can be expected that the same tendency would be shown even if the performance were evaluated using additional data. Thus, it can be concluded that learning a deep configuration or various kernels does not significantly affect performance. The reason for the test set accuracy tendency is the small size and simple features of the image. A single character image contains only one character, and therefore, high-quality images do not need to be used. The features in an image are simple as compared to those in other fields where CNNs are applied. In summary, the features of a single character image, which is small in size and consists of simple features, make it difficult to improve on the performance achieved by a deep configuration or various kernels of a neural network. And the model chosen was a model suitable for single character recognition. First, a stable loss trend model is considered. And since the test set performance of heavy models is better than shallow and light models, a reasonable choice seems to be a deep and heavy model. But it makes sense to adopt a shallow/light model. Because the test set performance of the two models is similar, there is no big difference in the loss value trend. And we need to choose a model that is less expensive to compute, taking into account the embedded system environment of the banknote counter. There are memory limits and fast processing requirements as shown in Table 4.

Finally, we compared the proposed CNN-based method with traditional pattern classification methods. For this purpose, the horizontal and vertical profiles of binarized character images were used as features, and labels were assigned to each character image; then, the test was conducted after training. A support vector machine (SVM) and a multi-layer perceptron (MLP) model were used for pattern classification. The learning result of the SVM method used a radial basis function (RBF) as the optimal kernel. The experimental results are shown in Table 5.

Table 5. Test accuracy comparison of proposed method and traditional pattern recognition methods (errors/trials)

Method	Test accuracy (%)
Proposed method with shallow/light-augmented case	99.92 (1/1290)
SVM with RBF kernel	94.65 (69/1290)
MLP	97.44 (33/1290)

Because SVM has advantages in binary classification, it is designed to be multi-classified in a hierarchical structure. However, although RBF is found to be an optimal kernel, it shows a relatively low accuracy rate. The MLP method achieved a better performance than the SVM method but showed limitations as a classification method after the features had been detected. It was confirmed that the features selected through learning in the CNN were optimized for character recognition.

5. Conclusion

In this paper, we proposed a character extraction method based on the aspect ratio of banknotes and a character recognition method based on a CNN. To obtain the image of each character, the banknote image was binarized and then distortion removal was performed through affine change, and the serial number ROI was extracted using the aspect ratio of the banknote. Then, a single character image was extracted from the serial number ROI. We designed four types of CNN-based neural networks for character recognition and selected that which was most suitable. The selection of the appropriate neural network was based on the loss value and character recognition performance trends. Our analysis of the loss value trend showed that the model with a large number of kernels in a deep configuration and the model with a small number of kernels in a shallow configuration were relatively small and stable. Further, a 100% recognition performance rate was shown for the former and a 99.92% recognition performance rate for the latter. The two performance rates are nearly the same. However, it was observed that a deep configuration or a large number of kernels was not suitable for single text image data, and thus, of the two models, which are similar in terms of the loss value and recognition performance, the model with the shallow configuration and a relatively small number of kernels is suitable for an embedded system environment. Data augmentation was applied during the experiments, which improved the performance in terms of the loss value and recognition performance trends.

References

- [1] J. Qian, D. Qian, and M. Zhang, "A digit recognition system for paper currency identification based on virtual instruments," in *Proceedings of the International Conference on Information and Automation*, Shandong, China, 2006, pp. 228-233.
- [2] P. S. Reel, G. Krishan, and S. Kotwal, "Image processing based heuristic analysis for enhanced currency recognition," *International Journal of Advancements in Technology*, vol. 2, no. 1, pp. 82-89, 2011.
- [3] M. Bhansali and P. Kumar, "An alternative method for facilitating cheque clearance using smart phones application," *International Journal of Application or Innovation in Engineering & Management*, vol. 2, no. 1, pp. 211-217, 2013.
- [4] M. T. Qadri and M. Asif, "Automatic number plate recognition system for vehicle identification using optical character recognition," in *Proceedings of the International Conference on Education Technology and Computer*, Singapore, 2009, pp. 335-338.
- [5] H. Penz, I. Bajla, A. Vrabl, W. Krattenthaler, and K. Mayer, "Fast real-time recognition and quality inspection of printed characters via point correlation," in *Proceedings of SPIE 4303: Real-Time Imaging V. Bellingham, WA: International Society for Optics and Photonics*, 2001, pp. 127-137.
- [6] P. K. Sahoo, S. A. K. C. Soltani, and A. K. Wong, "A survey of thresholding techniques," *Computer Vision, Graphics, and Image Processing*, vol. 41, no. 2, pp. 233-260, 1988.
- [7] T. T. Zhao, J. Y. Zhao, R. R. Zheng, and L. L. Zhang, "Study on RMB number recognition based on genetic algorithm artificial neural network," in *Proceedings of the International Congress on Image and Signal Processing*, Yantai, China, 2010, pp. 1951-1955.
- [8] B. Y. Feng, M. Ren, X. Y. Zhang, and C. Y. Suen, "Automatic recognition of serial numbers in bank notes," *Pattern Recognition*, vol. 47, no. 8, pp. 2621-2634, 2014.
- [9] B. B. Chaudhuri and U. Pal, "A complete printed Bangla OCR system," *Pattern Recognition*, vol. 31, no. 5, pp. 531-549, 1998.

- [10] W. Li, W. Tian, X. Cao, and Z. Gao, "Application of support vector machine (SVM) on serial number identification of RMB," in *Proceedings of the World Congress on Intelligent Control and Automation*, Jinan, China, 2010, pp. 6262-6266.
- [11] F. Mohammad, J. Anarase, M. Shingote, and P. Ghanwat, "Optical character recognition implementation using pattern matching," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 2, pp. 2088-2090, 2014.
- [12] M. K. Sarker and M. K. Song, "Segmentation and recognition of Korean vehicle license plate characters based on the global threshold method and the cross-correlation matching algorithm," *Journal of Information Processing Systems*, vol. 12, no. 4, pp. 661-680, 2016.
- [13] J. Singh, G. Singh, and R. Singh, "Optimization of sentiment analysis using machine learning classifiers," *Human-centric Computing and Information Sciences*, vol. 7, article no. 32, 2017.
- [14] T. K. Mishra, B. Majhi, and R. Dash, "A contour descriptors-based generalized scheme for handwritten Odia numerals recognition," *Journal of Information Processing Systems*, vol. 13, no. 1, pp. 174-183, 2014.
- [15] J. Kaur and M. Aggarwal, "A LabVIEW approach to detect the theft of Indian currency notes," *International Journal of Advanced Research in Electronics and Communication Engineering*, vol. 2, no. 1, pp. 84-88, 2013.
- [16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097-1105, 2012.
- [17] L. Prechelt, "Early stopping-but when?" in *Neural Networks: Tricks of the Trade*. Heidelberg: Springer, 1998, pp. 55-69.



Unsoo Jang <https://orcid.org/0000-0002-4860-1855>

He received his B.S. and M.S. degrees in computer science from Sangmyung University, Seoul, South Korea in 2014 and 2016, respectively. Since March 2016, he has been a Ph.D. student in the Department of Computer Science at Sangmyung University. His research interests include image processing and artificial intelligence.



Kun Ha Suh <https://orcid.org/0000-0002-1500-3111>

He received his B.S. and M.S. degrees in computer science from Sangmyung University, Seoul, South Korea in 2015 and 2017, respectively. Since September 2017, he has been a Ph.D. student in the Department of Computer Science at Sangmyung University. His research interests include image processing, computer vision, and artificial intelligence.



Eui Chul Lee <https://orcid.org/0000-0001-6504-3333>

He received his B.S. degree in software from Sangmyung University, Seoul, South Korea, in 2005. He received his M.S. and Ph.D. degrees in computer science from Sangmyung University in 2007 and 2010, respectively. He was a researcher in the division of fusion and convergence of mathematical sciences at the National Institute for Mathematical Sciences from March 2010 to February 2012. Since March 2012, he has been a professor in the Department of Human Centered Artificial Intelligence at Sangmyung University, Seoul, South Korea. His research interests include computer vision, image processing, pattern recognition, and human-computer interfaces.