J Inf Process Syst, Vol.14, No.1, pp.205~217, February 2018
https://doi.org/10.3745/JIPS.04.0061

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Variations of AlexNet and GoogLeNet to Improve Korean Character Recognition Performance

Sang-Geol Lee*, Yunsick Sung**, Yeon-Gyu Kim***, and Eui-Young Cha***

Abstract
Deep learning using convolutional neural networks (CNNs) is being studied in various fields of image recognition and these studies show excellent performance. In this paper, we compare the performance of CNN architectures, KCR-AlexNet and KCR-GoogLeNet. The experimental data used in this paper is obtained from PHD08, a large-scale Korean character database. It has 2,187 samples of each Korean character with 2,350 Korean character classes for a total of 5,139,450 data samples. In the training results, KCR-AlexNet showed an accuracy of over 98% for the top-1 test and KCR-GoogLeNet showed an accuracy of over 99% for the top-1 test after the final training iteration. We made an additional Korean character dataset with fonts that were not in PHD08 to compare the classification success rate with commercial optical character recognition (OCR) programs and ensure the objectivity of the experiment. While the commercial OCR programs showed 66.95% to 83.16% classification success rates, KCR-AlexNet and KCR-GoogLeNet showed average classification success rates of 90.12% and 89.14%, respectively, which are higher than the commercial OCR programs' rates. Considering the time factor, KCR-AlexNet was faster than KCR-GoogLeNet when they were trained using PHD08; otherwise, KCR-GoogLeNet had a faster classification speed.

Keywords
Classification, CNN, Deep Learning, Korean Character Recognition

1. Introduction

Convolutional neural networks (CNNs) are beginning to receive considerable use in the image-recognition field [1–3], and their image-recognition performance is being enhanced in various situations. In addition, studies to enhance character-recognition performance are being conducted [4,5], and these studies show marked progress in Chinese character recognition. Zhang [6] compared the recognition accuracy of Chinese characters through several CNN architectures of different depths. Moreover, Zhong et al. [7] created a characteristic map according to the directional feature. This newly created map was used with existing training data and showed advanced recognition accuracy. Yang et al. [8] used Chinese character domain information and applied it to an experiment with CNNs.

However, studies applying CNNs to Korean character recognition have not been conducted in earnest. For the last three decades, studies on various methods have advanced Korean character

Manuscript received May 12, 2017; first revision June 23, 2017; accepted July 31, 2017.
Corresponding Author: Eui-Young Cha (eycha@pusan.ac.kr), Yeon-Gyu Kim (dusrb2003@naver.com)
* Human Resources Development Group for Women Maker in Integrated Engineering, Dongguk University, Seoul, Korea (sglee@dongguk.edu)
** Dept. of Multimedia Engineering, Dongguk University, Seoul, Korea (sung@dongguk.edu)
*** Dept. of Computer Science Engineering, Pusan National University, Busan, Korea (dusrb2003@naver.com)

recognition [9–14]; however, this is considered a slow development speed when compared with the recognition rate of other characters using CNNs. Recently, Kim and Xie [15] conducted a study on Korean character recognition using a CNN; however, their experimental data was limited and the CNN configuration was relatively simple.

In this study, a large-scale Korean character database, PHD08 [16], is used for the experiment. PHD08 is the largest Korean character database currently in existence, and it has a total of 5,139,450 characters. PHD08 is very useful for training because every character has different fonts, resolutions, rotations, and noise levels. After training newly designed CNNs, Korean character recognition (KCR)-AlexNet and KCR-GoogLeNet, with divided training and test data from PHD08 at a constant rate, we compare the test accuracy depending on the training iterations. After demonstrating which CNN network has higher test accuracy, we present a classification experiment, which is based on additional Korean character data with fonts that are not in PHD08, to ensure the objectivity of the experiment. Then, we compare the classification success rate with online commercial optical character recognition (OCR) programs to compare the pros and cons of KCR-AlexNet and KCR-GoogLeNet. Moreover, we measured the time to conduct each experiment and used it as one of the factors for evaluating the performance between KCR-AlexNet and KCR-GoogLeNet.

The structure of this paper is as follows: Section 2 gives a brief introduction to CNNs and describes our KCR-AlexNet and KCR-GoogLeNet architectures in detail. Our experiment is discussed in Section 3. Section 3.1 describes the process of training a CNN network, from organizing the experimental data to drawing a test-accuracy curve graph, with KCR-AlexNet and KCR-GoogLeNet. Section 3.2 shows an experiment classifying Korean characters with fonts that are not in PHD08, and compares the classification performance of KCR-AlexNet, KCR-GoogLeNet, and other commercial programs. Conclusions are drawn in Section 4.

2. CNN Architecture Design for Korean Character Recognition

2.1 Introduction to Convolutional Neural Network

The CNN that extracts local feature information from the input data used for training is began to use from LeNet, which was designed by LeCun et al. [17] to recognize digit images. CNN is a type of neural network that uses convolution, pooling, inner-products, etc., repeatedly, and has become the most frequently used architecture for deep learning in the image-recognition field. Recently, the problem of over-fitting has been solved using the rectified linear unit (ReLU) non-linear activation function and a drop-out layer. When the data enters the CNN's convolutional layer, the output value is calculated as in (1) and propagated to a node in the next layer.

$$\mathrm{x}_\mathrm{i}^\mathrm{l} = \sum \mathrm{x}^{\mathrm{l}\text{-}1} \times \mathrm{k}_\mathrm{i}^\mathrm{l} + \mathrm{b}_\mathrm{i}^\mathrm{l} \tag{1}$$

where x_i^l is the result of calculating the i^{th} node of the l^{th} layer. It is calculated by accumulating the results of multiplying the node value at the $(l-1)^{\text{th}}$ layer by the i^{th} kernel map of the l^{th} layer, and adding the i^{th} bias at the l^{th} layer. For the results of the final CNN layer, the term Loss or Error is used to indicate how close the output value is to the target value. The average loss over all |D| instances throughout dataset D is calculated as follows.

$$L(W) = \frac{1}{|D|}\sum_{i}^{|D|} f_w\left(X^{(i)}\right) + \lambda r(W) \tag{2}$$

where W is the weight-parameter map of the current network, $f_w(X^{(i)})$ is the loss on data instance $X^{(i)}$, and $r(W)$ is a regularization term with constant value λ. When the loss calculation is completed by (2), the weight-parameter map must be updated in every training iteration, as in (3) and (4). This equation means that we follow the SGD (Stochastic Gradient Descent) algorithm to minimize the value of the Loss, which is the goal of training.

$$V_{t+1} = \mu V_t - \alpha \nabla L(W_t) \tag{3}$$

$$W_{t+1} = W_t + V_{t+1} \tag{4}$$

Firstly, W_{t+1} means the updated weight-parameter map at training iteration $t+1$, which is calculated by update value V_{t+1}. We use constant value μ, called momentum, to calculate V_{t+1}; V_t means the update value at the previous training iteration. Lastly, α means the learning rate for the current training iteration and $\nabla L(W_t)$ means the negative loss gradient from the weight-parameter map at the previous training iteration. Specific constant values, like α or μ, will be presented in Section 3.

2.2 Design of Two CNN Architectures for Korean Character Recognition

2.2.1 KCR-AlexNet

The CNN architecture that received the most attention after LeNet was AlexNet by Krizhevsky et al. [18]. AlexNet won first place at the Imagenet Large-Scale Visual Recognition Challenge 2012 (ILSVRC-2012) [19] contest. The overall architecture of KCR-AlexNet is the same as AlexNet, but KCR-AlexNet uses a 56×56-pixel input data size for Korean character images, which is smaller than AlexNet's input data size of 256×256 for natural images. In addition, while the output layer of the existing AlexNet only has 1,000 nodes for classifying ILSVRC's classes, KCR-AlexNet needs 2,350 nodes at the output layer to classify PHD08's 2,350 Korean character classes.

The details of KCR-AlexNet are depicted in Fig. 1; it consists of five convolutional layers, three max pooling layers, and three fully connected layers. The ReLU non-linearity activation function is applied to each convolutional and fully connected layer in KCR-AlexNet, and the final output value is calculated by Softmax.

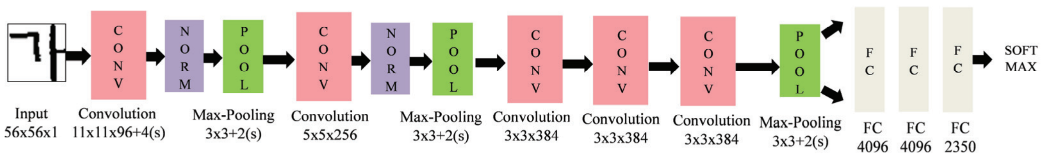

Fig. 1. KCR-AlexNet architecture.

2.2.2 KCR-GoogLeNet

We designed another CNN architecture, KCR-GoogLeNet, based on GoogLeNet developed by Szegedy et al. [20]. CNN-based GoogLeNet won a recognition field in ILSVRC-2014, and has a much deeper architecture than existing CNN architectures. KCR-GoogLeNet has 22 layers of weight parameters, which are twice KCR-AlexNet's 8 layers of weight parameters. However, the biggest feature of GoogLeNet and KCR-GoogLeNet is that the inception module is included in the architecture. Szegedy et al. configured the inception module using the method of Arora et al. [21]. The inception module is designed as a network structure in a network; and it is different from a conventional CNN in that it has only a one-dimensional series configuration. The same inception modules are used in GoogLeNet and KCR-GoogLeNet, and are shown in Fig. 2 with a depth of two. The depth indicates how many layers with a weight parameter are connected.

The inception module is a method introduced to effectively express the features of the local space. After subdividing the regional characteristics of the kernel space into sizes 1×1, 3×3, and 5×5 to calculate the convolutional value, all convolutional result values are concatenated in the last layer of the inception module. The 1×1 convolutional layer applying the ReLU activation function is used to reduce the complexity of the calculations occurring in the 3×3 and 5×5 convolutional layers. The biggest difference between GoogLeNet and KCR-GoogLeNet is that GoogLeNet uses nine inception modules and KCR-GoogLeNet uses only three inception modules. This is because GoogLeNet's purpose is to classify the nature of a 256×256×3 image size and KCR-GoogLeNet's purpose is to classify small Korean characters of size 56×56×1. The KCR-GoogLeNet architecture is shown in Fig. 3 and the detail size of each layer, including the inception modules, is introduced in Table 1.

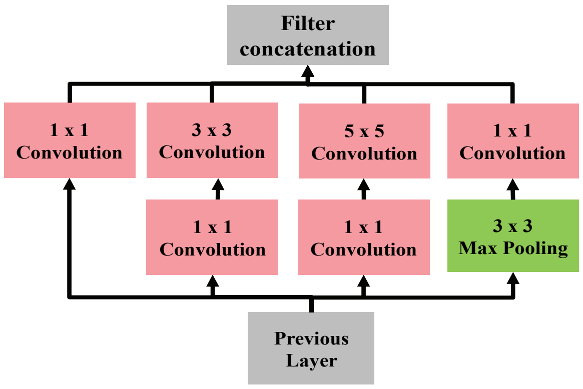

Fig. 2. Inception module for GoogLeNet and KCR-GoogLeNet.

Table 1. KCR-GoogLeNet incarnation of the inception architecture

Type (patch size / stride)	Output size	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	Pool project
Input (56×56×1)							
Convolution (3×3 / 1)	28×28×192		64	192			
Max-Pool (3×3 / 2)	14×14×192						
Inception(a)	14×14×256	64	96	128	16	32	32
Inception(b)	14×14×480	128	128	192	32	96	64
Max-Pool (3×3 / 2)	7×7×480						
Inception(c)	7×7×512	192	96	208	16	48	64
Average- Pool (7×7 / 1)	1×1×512						
Convolution (1×1 / 1)	1×1×128						
Fully-Connected	1×1×1024						
Dropout (70%)	1×1×1024						
Fully-Connected	1×1×2350						
Softmax	1×1×2350						

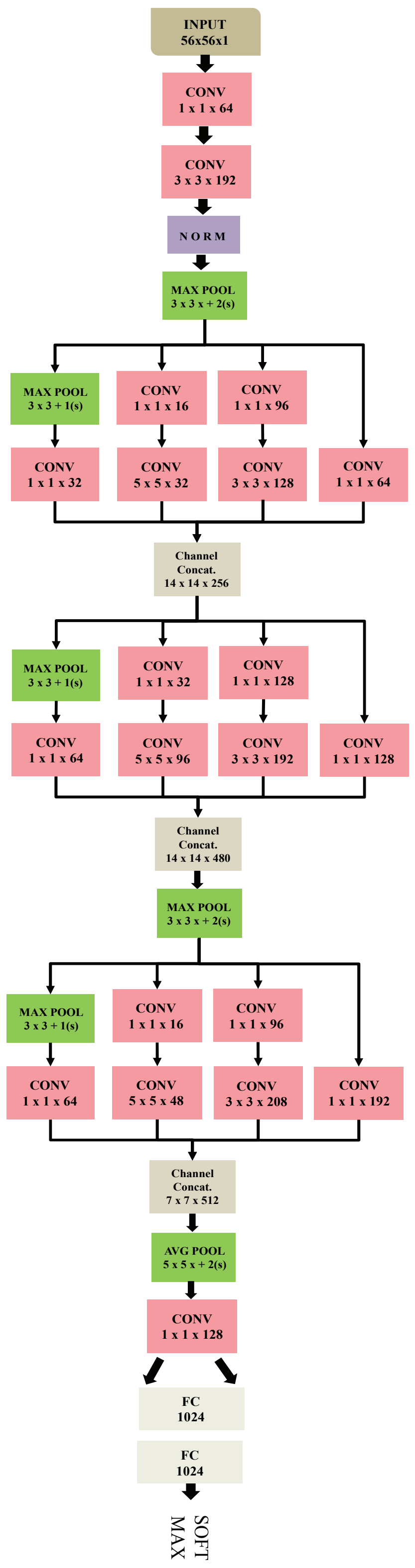

Fig. 3. KCR-GoogLeNet architecture.

3. Experiments Results

In this section, we describe and analyze the results for the two experiments. Firstly, in Section 3.1, after training KCR-AlexNet and KCR-GoogLeNet on PHD08, we compared the test accuracy in accordance with the training iteration and the time it takes for a training iteration. Secondly, in Section 3.2, after entering new Korean character data with fonts that are not in PHD08 into KCR-AlexNet and KCR-GoogLeNet, we compared their classification performance and classification time. Additionally, we compared the classification performance with other online commercial OCR-programs to ensure the objectivity of the experiments.

3.1 Experiments for PHD08

3.1.1 Experimental data

PHD08 is the Korean character database used for experiments. The scanned images are saved in binary form after printing Korean characters created in a variety of conditions. With 2,187 samples of each character, which is the Korean Standard (KS) completion of 2,350 Hangul characters, all samples have a different font, size, rotation, noise level, etc., as shown in Table 2. The experiments change the binary data of PHD08 to binary images to be used as input data for KCR-AlexNet and KCR-GoogLeNet. However, because of the constraint that all CNN input data must be the same size, we changed all of the data sizes to 56×56. Linear interpolation [22] was used to change the size, and an example of the final changed Korean character input data is shown in Fig. 4.

Table 2. PHD08 composition

Characters	Samples	Type	Fonts	Size (pt)	Rotation	Noise
2,350	2,187	Binary	9	12, 13, 14	$-3^{\circ}, 0^{\circ}, 3^{\circ}$	3-level

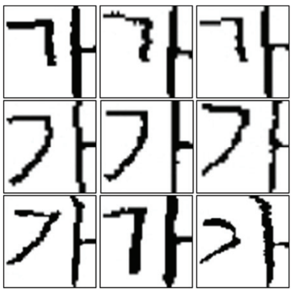

Fig. 4. Example of transformed Input data from PHD08.

In this paper, to investigate the difference in test accuracy depending on the training iterations when the number of training data is large or small, we constructed five experimental data sets, as shown in Table 3. A total of 5,139,450 PHD08 data elements were divided into training and testing data, with ratios of 1:0.5 to 1:8 composing the five sets. While KCR-AlexNet and KCR-GoogLeNet train each data set, we confirm that the difference in test accuracy depends on the training iteration and shows which network is better for training Korean characters.

3.1.2 Experiment environment

Some of the parameters that were used in common by KCR-AlexNet and KCR-GoogLeNet were given equal initial values. SGD, a method for modifying the weight parameter that has the most critical role in the training, was explained in Section 2. The initial learning rate (α) used 0.9 and the momentum constant (μ) used 0.01 for all training experiments. In addition, the learning rate was decreased 0.96-fold for every 10,000 training iteration. Batch sizes of 56 and 50 were used for the training phase and test phase, respectively; the batch size is the number of images used as input data for each training and test iteration. Throughout this experiment, a parallel-computing graphics card GTX-970 (CUDA v7.0) was used for quick operations, and we conducted the test on a public framework, Caffe [23].

Table 3. Five experimental data sets

Set	Ratio	Training	Testing
E1-Set_1	1:0.5	1,458 × 2,350 = 3,426,300	729 × 2,350 = 1,713,150
E1-Set_2	1:1	1,093 × 2,350 = 2,568,550	1,094 × 2,350 = 2,570,900
E1-Set_3	1:2	729 × 2,350 = 1,713,150	1,458 × 2,350 = 3,426,300
E1-Set_4	1:4	437 × 2,350 = 1,026,950	1,750 × 2,350 = 4,112,500
E1-Set_5	1:8	243 × 2,350 = 571,050	1,944 × 2,350 = 4,568,400

3.1.3 Experiment analysis

We show that the test-accuracy curve graph depends on the training iteration in Fig. 5, while KCR-AlexNet and KCR-GoogLeNet are training the data, E1-Set_1 to E1-Set_5 in Table 3. In the results of the experiment, KCR-AlexNet and KCR-GoogLeNet always converged at over 98% test accuracy when training each of the five data sets. The exact values of the test accuracy at the end of the training are shown in Table 4. If sufficient training continues, even if the proportion of the training data is relatively small, like E1–Set_5, the accuracy of the test can increase at any time.

In the case of top-1 as shown in Table 4, KCR-GoogLeNet has higher test accuracy than KCR-AlexNet for all of the data sets. However, according to Fig. 5, we can see that the test accuracy of KCR-GoogLeNet always converges in the training iteration, which is more progress than occurs in the training iterations of KCR-AlexNet. This is because the KCR-GoogLeNet architecture spends more time in the inception module, finding features of a small compact area, than KCR-AlexNet. Such characteristics can greatly affect the training time, which is an important factor when designing a CNN. Therefore, when the two networks continued with training, we measured the average time spent on each training iteration and recorded it in Table 4. KCR-AlexNet took 0.042 seconds and KCR-GoogLeNet took 0.424 seconds on average to work out each of the single iterations. If the time required for training is a significant constraint, KCR-AlexNet is more effective for training Korean characters.

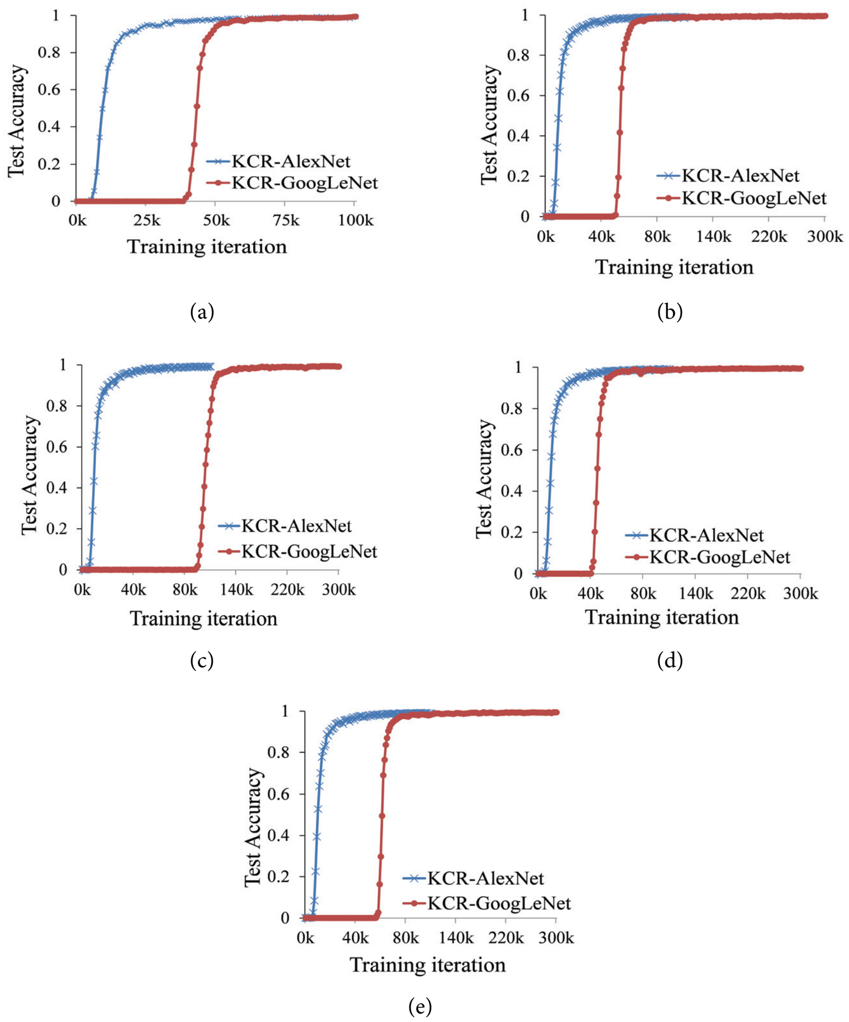

Fig. 5. Comparison between KCR-AlexNet and KCR-GoogLeNet. (a) E1-Set_1, (b) E1-Set_2, (c) E1-Set_3, (d) E1-Set_4, and (e) E1-Set_5.

Table 4. Test accuracies for the last training iteration and average times for a single iteration

Set	Network	Top-1 (%)	Top-2 (%)	Top-5 (%)	Time (s)
E1-Set_1	KCR-AlexNet	99.31	99.85	99.92	0.041
	KCR-GoogLeNet	99.41	99.73	99.89	0.433
E1-Set_2	KCR-AlexNet	99.00	99.78	99.91	0.039
	KCR-GoogLeNet	99.56	99.86	99.93	0.428
E1-Set_3	KCR-AlexNet	98.98	99.75	99.90	0.044
	KCR-GoogLeNet	99.14	99.74	99.90	0.421
E1-Set_4	KCR-AlexNet	98.16	99.83	99.93	0.044
	KCR-GoogLeNet	99.66	99.88	99,94	0.425
E1-Set_5	KCR-AlexNet	98.91	99.79	99.91	0.040
	KCR-GoogLeNet	99.40	99.83	99.90	0.411

3.2 Experiment about Classification with Other Applications

This experiment compares the time required for classification and the classification success rate between KCR-AlexNet and KCR-GoogLeNet on the new Korean character data with fonts that are not in PHD08. In addition, we compared the classification success rates with commercial OCR programs ABBYY FineReader 12 [24], ABC-OCR [25], and Office Lens [26] to ensure the objectivity of the experiment.

3.2.1 Experimental data

We used the characters in the Korean national anthem as experimental data. The national anthem is composed of four verses and a chorus. Each verse has 28 characters, while the chorus has 24 characters, so 136 characters are in the anthem. However, we do not count duplicate characters; thus, only 82 characters are used for this experiment.

Table 5. Used fonts for PHD08 and new data set

PHD08	Example of forms	New data set name	Example of forms
Bada	Test for this fonts	E2–Set_1 (Gulim)	Test for this fonts
Dotum	Test for this fonts	E2–Set_2 (Gungseo)	Test for this fonts
Gothic	Test for this fonts	E2–Set_3 (Batang)	Test for this fonts
Hanyanghaeseo	Test for this fonts	E2–Set_4 (HYMokpanL)	Test for this fonts
Headline	Test for this fonts	E2–Set_5 (OI)	Test for this fonts
Myungjo	Test for this fonts	E2–Set_6 (YangjaeinChangjaeM)	Test for this fonts
Namu	Test for this fonts	E2–Set_7 (MDgaeseongchae)	Test for this fonts
Saemmul	Test for this fonts	E2–Set_8 (Humanpyeonjichae)	Test for this fonts
YeopseoHangul	Test for this fonts	E2–Set_9 (Ganeun Ansangsoochae)	Test for this fonts
-	-	E2–Set_10 (HYcrystalM)	Test for this fonts

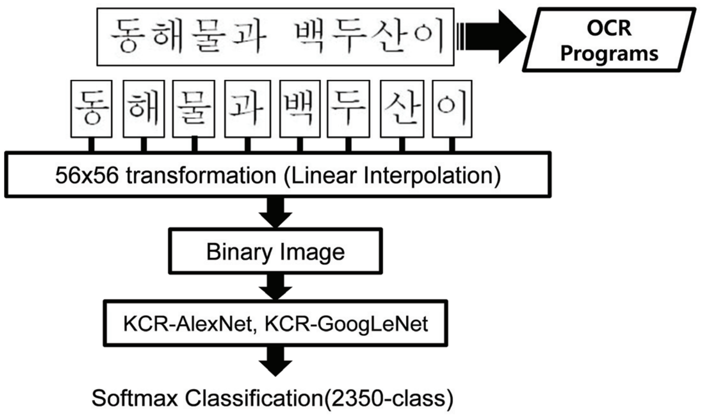
동해물과 백두산이
OCR Programs
동
해
물
과
백
두
산
이
56x56 transformation (Linear Interpolation)
Binary Image
KCR-AlexNet, KCR-GoogLeNet
Softmax Classification(2350-class)

Fig. 6. Comparison between KCR-AlexNet and KCR-GoogLeNet.

We made 10 data sets as shown in Table 5. Each set has all 82 characters, for a total of 820 characters. All fonts used in new data sets are not PHD08 fonts. However, in this experiment, new data sets were used as input data for the classification after the procedure, as in Fig. 6. After selecting the specified area of each character from the image, the width and height are transformed to the size of 56×56 and converted to a binary image. In addition, we used a weight-parameter map for classification from the trained KCR-AlexNet and KCR-GoogLeNet with E1–Set_1 in Section 3.1.

3.2.2 Experiment analysis

Table 6 shows the classification success rate for KCR-AlexNet, KCR-GoogLeNet, and other OCR programs. For the 820 distinct Korean characters used in the experiment, three applications, ABBYY, ABC-OCR, and Office Lens, respectively showed 72.07%, 83.17%, and 66.95% for classification success rates. On the other hand, KCR-AlexNet and KCR-GoogLeNet showed classification success rates of 90.12% and 89.14%, respectively, which is better performance than the existing programs. However, all networks and programs used in the experiment showed relatively low success rates for forms similar to human handwriting, e.g., "OI (E2–Set_5)," "Humanpyeonjichae (E2–Set_8)," and "Ganeun Ansangsoochae (E2–Set_9)." This phenomenon occurred because the handwritten font patterns and the PHD08 font patterns did not match. Thus, additional training data is needed to increase the classification success rate for handwritten pattern data.

Table 6. Classification success rate comparison between KCR-AlexNet, KCR-GoogLeNet and other programs

Set	ABBYY	ABC-OCR	Office Lens	KCR-AlexNet	KCR-GoogLeNet	Avg. (Top-1) (%)
E2–Set_1	80/82	82/82	78/82	Top-1 (79/82) Top-2 (81/82)	Top-1 (80/82) Top-2 (82/82)	97.31
E2–Set_2	80/82	60/82	76/82	Top-1 (82/82) Top-2 (82/82)	Top-1 (82/82) Top-2 (82/82)	92.68
E2–Set_3	82/82	81/82	82/82	Top-1 (73/82) Top-2 (80/82)	Top-1 (79/82) Top-2 (82/82)	96.82
E2–Set_4	74/82	77/82	69/82	Top-1 (78/82) Top-2 (80/82)	Top-1 (73/82) Top-2 (76/82)	90.48
E2–Set_5	42/82	61/82	N/A*	Top-1 (70/82) Top-2 (73/82)	Top-1 (64/82) Top-2 (72/82)	57.8
E2–Set_6	76/82	77/82	76/82	Top-1 (80/82) Top-2 (82/82)	Top-1 (82/82) Top-2 (82/82)	95.36
E2–Set_7	50/82	71/82	60/82	Top-1 (70/82) Top-2 (75/82)	Top-1 (77/82) Top-2 (77/82)	80.00
E2–Set_8	30/82	52/82	33/82	Top-1 (70/82) Top-2 (72/82)	Top-1 (64/82) Top-2 (69/82)	60.73
E2–Set_9	N/A	40/82	N/A	Top-1 (62/82) Top-2 (67/82)	Top-1 (58/82) Top-2 (66/82)	39.02
E2–Set_10	77/82	81/82	75/82	Top-1 (75/82) Top-2 (81/82)	Top-1 (72/82) Top-2 (78/82)	92.68
Avg. (Top-1) (%)	72.07	83.17	66.95	90.12	89.14	80.29

Comparing only the performance of KCR-AlexNet and KCR-GoogLeNet, while KCR-GoogLeNet had higher test accuracy for training PHD08 in Section 3.1, this experiment showed a slightly higher classification success rate for KCR-AlexNet. As additional experimental results, KCR-AlexNet took between 0.027 seconds and 0.036 seconds to classify one character, and KCR-GoogLeNet took between 0.021 seconds and 0.025 seconds. Such a difference may be significant when classifying bulk characters. Therefore, the CNN should be selected in view of the classification time and the classification success rate, depending on the situation.

4. Conclusions

CNN structures, KCR-AlexNet and KCR-GoogLeNet, used for Korean character recognition in this paper showed more than 98% test accuracy for PHD08. Further, for an objective evaluation of this paper, we generated new Korean character data with fonts that did not exist in PHD08 and compared the performance with commercial OCR programs that are present online. The experimental results showed that the classification success rates of KCR-AlexNet and KCR-GoogLeNet were higher than the success rates of existing OCR programs, which proved the classification performance for Korean characters with various fonts.

However, additional discussion was required for the performance comparison between KCR-AlexNet and KCR-GoogLeNet. If the test used PHD08, the test accuracy of the KCR-GoogLeNet was higher; however, the classification success rate of KCR-AlexNet was higher in the experiments classifying newly created Korean character data.

In addition to the test accuracy and classification success rate, we measured the temporal component for the experiments. While the training time for PHD08 was short for KCR-AlexNet, the time required to classify one character was short for KCR-GoogLeNet. Thus, we showed that the training time on a given database, the test accuracy from a network, the classification success rate, and the time required for classification must be considered when choosing a CNN for recognizing Korean characters.

Acknowledgement

This work was supported by a 2-Year Research Grant of Pusan National University.

References

[1] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Networks*, vol. 32, pp. 333-338, 2012.

[2] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," 2014 [Online]. Available: https://arxiv.org/abs/1404.2188.

[3] P. L. Callet, C. Viard-Gaudin, and D. Barba, "A convolutional neural network approach for objective video quality assessment," *IEEE Transactions on Neural Networks*, vol. 17, no. 5, pp. 1316-1327, 2006.

[4] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Convolutional neural network committees for handwritten character classification," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR),* Beijing, China, 2011, pp. 1135-1139.

[5] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, “End-to-end text recognition with convolutional neural networks,” in *Proceedings of the 21st International Conference on Pattern Recognition(ICPR 2012)*, Tsukuba, Japan, 2012, pp. 3304-3308.

[6] Y. Zhang, “Deep convolutional network for handwritten Chinese character recognition,” [Online]. Available: http://yuhao.im/files/Zhang_CNNChar.pdf.

[7] Z. Zhong, L. Jin, and Z. Xie, “High performance offline handwritten Chinese character recognition using GoogLeNet and directional feature map,” in *Proceedings of the 13th International Conference on Document Analysis and Recognition (ICDAR)*, Nancy, France, 2015, pp. 846-850.

[8] W. Yang, L. Jin, Z. Xie, and Z. Feng, “Improved deep convolutional neural network for online handwritten Chinese character recognition using domain-specific knowledge,” in *Proceedings of the 13th International Conference on Document Analysis and Recognition (ICDAR)*, Nancy, France, 2015, pp. 551-555.

[9] D. C. Hwang and S. S. Kim, “Hangul recognition using path following algorithm,” *IE Interfaces*, vol. 3, no. 2, pp. 53-62, 1990.

[10] B. K. Sin, and J. H. Kim, “On-line handwritten character recognition with hidden Markov models,” in *Proceedings of the 4th Annual Conference on Human and Cognitive Language Technology,* Seoul, Korea, 1992, pp. 533-542.

[11] J. K. Chung, S. I. Kim, and J. C. Namgung, “A study on an on-line handwritten Hangul character recognition by identifying relative positions of strokes,” *Journal of Information Technology Applications and Management*, vol. 4, no. 2, pp. 65-78, 1997.

[12] J. Y. Ha, and B. K. Shin, “Optimization of number of states in HMM for on-line Hangul recognition,” *Proceeding of the Korea Information Science Society*, vol. 25, no. 2, pp. 372-374, 1998.

[13] J. H. Lee, J. H. Ahn, and I. B. Lee, “Elastic curvature matching for online handwritten Hangul recognition,” *Proceeding of the Korea Information Science Society*, vol. 35, no. 2, pp. 238-239, 2008.

[14] H. S. Cho, “A new feature-extraction method using wavelet transformation and fuzzy data for handwritten Hangul recognition,” *Journal of Korean Institute of Information Technology*, vol. 3, no. 4, pp. 11-17, 2005.

[15] I. J. Kim and X. Xie, “Handwritten Hangul recognition using deep convolutional neural network,” *International Journal of Document Analysis and Recognition*, vol. 18, no. 1, pp. 1-13, 2011.

[16] D. S. Ham, D. Y. Lee, I. S. Jung, and I. S. Oh, “Construction of printed Hangul character database PHD08,” *Journal of the Korea Contents Association*, vol. 8, no. 11, pp. 33-40, 2008.

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceeding of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.

[18] A. Krizhevsky, I. Sutskever and G. E. Hinton, “ImageNet classification with deep convolutional neural network,” in *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS'12)*, Lake Tahoe, NV, 2012, pp. 1097-1105.

[19] ImageNet Large Scale Visual Recognition Challenge [Online]. Available: http://www.image-net.org/challenges/LSVRC/.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceeding of the IEEE Conference on Computer Vison and Patter Recognition*, Boston, MA, 2015, pp. 1-9.

[21] S. Arora, A. Bhaskara, R. Ge, and T. Ma, “Provable bounds for learning some deep representations,” 2013 [Online]. Available: https://arxiv.org/abs/1310.6343.

[22] Linear interpolation [Online]. Available: https://en.wikipedia.org/wiki/Linear_interpolation.

[23] Caffe (convolutional architecture for fast feature embedding) [Online]. Available: http://caffe.berkeleyvision.org/.

[24] ABBYY FineReader 12 [Online]. Available: http://www.retia.co.kr/cnt/products/products.html?category=1&uid=24&name=finereader-12&tab=1.

[25] ABC OCR scanner app [Online]. Available: https://itunes.apple.com/us/app/scanner-ocr-optical-character/id777913435.
[26] Office Lens app [Online]. Available: https://itunes.apple.com/kr/app/office-lens/id975925059?mt=8.

Sang-Geol Lee https://orcid.org/0000-0002-5771-0834

He received the B.S. degree in Computer Science and Engineering from Pusan National University, Busan, Korea, in 2003 and the M.S. degree in Computer Engineering from Pusan National University, Busan, Korea, in 2005. He received the Ph.D. degree in Electrical and Computer Engineering from Pusan National University, Busan, Korea, in 2016. His research interests include image processing, signal processing, pattern recognition, neural networks, and machine learning.

Yunsick Sung https://orcid.org/0000-0003-3732-5346

He is currently an Assistant Professor in the department of Multimedia Engineering at Dongguk University, Seoul, Korea. He received the B.S. degree in Division of Electrical and Computer Engineering from Pusan National University, Busan, Korea, in 2004, the M.S. degree in Computer Engineering from Dongguk University, Seoul, Korea, in 2006, and the Ph.D. degree in Game Engineering from Dongguk University, Seoul, Korea, in 2012. He was employed as a Member of the Researcher at Samsung Electronics in Korea, between 2006 and 2009. He was the Plural Professor at Shinheung College, Gyeonggi-do, Korea, in 2009, and at Dongguk University, Seoul, Korea, in 2010. He was also the postdoctoral fellow at University of Florida, FL, USA, between 2012 and 2013. His research interests are focused on the areas of games, pervasive computing, and robotics.

Yeon-Gyu Kim https://orcid.org/0000-0002-5627-1032

He received the B.S. degree in Computer Science and Engineering from Pusan National University, Korea in 2015, the M.S. degree in Electrical and Computer Engineering from Pusan National University, Korea, in 2017.

Eui-Young Cha https://orcid.org/0000-0002-7531-3242

He received the B.S. degree in Electronic Engineering from Kyungpook National University, Korea, in 1979, the M.S. degree in Computer Science from Seoul National University, Korea, in 1982, the Ph.D. degree in Computer Engineering from Seoul National University, Seoul, Korea, in 1998. He was a researcher of ETRI (Electronics and Telecommunications Research Institute), Korea from 1981 to 1985. He was a Visiting Professor with the Faculty of Engineering, University of London, UK from 1995 to 1996. He is currently a professor in Department of Computer Science and Engineering, Pusan National University, Korea.