2017 14th IAPR International Conference on Document Analysis and Recognition

# Extremely Sparse Deep Learning using Inception Modules with Dropfilters

Woo-Young Kang
Department of Computer Science and Engineering
Seoul National University &
Surromind Robotics
Seoul 151-742, Republic of Korea
wykang@bi.snu.ac.kr

Kyung-Wha Park
Brain Science Program
Seoul National University
Seoul 151-742, Republic of Korea
kwpark@bi.snu.ac.kr

Byoung-Tak Zhang
Department of Computer Science and Engineering,
Brain Science Program
Seoul National University &
Surromind Robotics
Seoul 151-742, Republic of Korea
btzhang@bi.snu.ac.kr

***Abstract*—This paper reports a successful application of highly sparse convolutional network model for offline handwritten character recognition. The model makes use of spatial dropout techniques named dropfilters for sparsifying the inception modules in GoogLeNet, resulting in extremely sparse deep networks. The model is industry-deployable regarding model size and performance, which trained by a handwritten dataset of 520 classes and 260,000 Hangul(Korean) characters for tablet PCs and smartphones. The proposed model obtained significant improvement in recognition performance while the number of parameters is much smaller than that of the LeNet, a classical sparse convolutional network. We also evaluated the dropfiltered inception networks on the handwritten Hangul dataset and achieved 3.275% higher recognition accuracy with approximately three times fewer parameters than a deep network based on LeNet structure without dropfilters.**

Fig. 1. Examples of handwritten Hangul data

## I. Introduction

In the field of handwritten character recognition, the demand for automated recognition of postal codes and topic classification of documents has tremendously increased, and full-scale research of the handwritten character recognition began actively from the 1990s. In the past, models based on the probabilistic graphical model [1] or the support vector machine [2] were generally used for handwritten recognition. Recently, models based on the Convolutional Neural Networks(CNN) have improved image recognition performance [3]. So, other studies using CNN have also been proposed in the field of handwritten digits and alphabets recognition [4] [5]. However, Hangul has more complicated structure than digits and alphabets as depicted in Fig. 1. Furthermore, Hangul has more than 2,000 characters, which is more than the Latin alphabet. Hence the use of CNNs is indispensable and preferable to recognize the handwritten Hangul. According to this trend of handwritten recognition, the CNN-based research on Handwritten Hangul Recognition(HHR) has recently been actively studied [6], [7]. Since the handwritten recognition module is a default application under resource-constrained environments(e.g. smartphone, tablet PC, etc.), it is prudent to jointly consider performance metrics like computational speed and recognition accuracy for the recognition tasks.

The inception module used in GoogLeNet is known to have impressive recognition accuracy with fewer parameters [8]. Because of these advantages, it is suitable for the model to be used in handwritten character recognition. Further, Kang et al. applied the inception module to the HHR and had higher recognition accuracy with much fewer parameters [9]. However, a deep learning model can easily overfit to the data in general if we stack many convolutional layers. So, it is also important to lower the generalization error using regularization techniques to mitigate the overfitting problem.

The dropout technique is a method to address that overfitting problem [10]. The dropout technique is widely used in the artificial neural networks because it removes hidden neurons arbitrarily to prevent co-adaptation between neurons and obtain ensemble learning effect of sub-models. Further, by using the dropout, activations of hidden layers become sparse without other regularizers for sparsity such as weight decay or sparsity penalty proposed at [11]. So, we can get sparse feature representations using the dropout, which is helpful to reduce generalization error. However, applying this technique to the feature map of convolutional layers of a CNN was not very successful [12]. In this paper, we propose a regularization method named dropfilters, which is a modified dropout technique for dropping the convolutional masks with

2379-2140/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDAR.2017.80

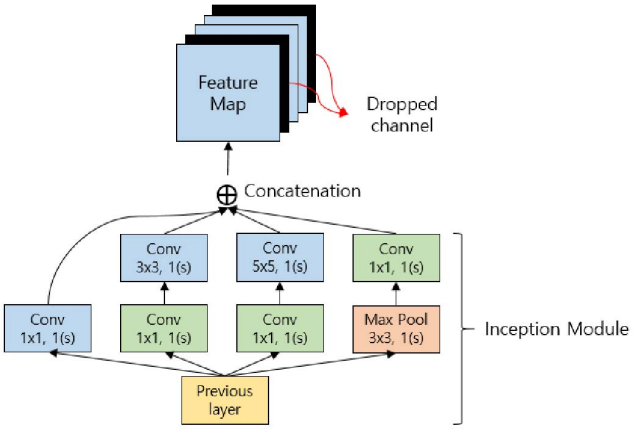


Fig. 2. Concept of the dropfilters with the inception module. The $\oplus$ mark indicates concatenation, and the black shaded square on the feature map indicates the dropped channel.

probability $p$. We briefly describe a core part of our model in Fig. 2. The dropfilters results in the same operation as spatial dropout [13], but we consider the spatial dropout as an ensemble learning of sub-networks at Section 4.

An experiment with CIFAR-10 data [14] was performed to verify the effect of the dropfilters. Also, we observed when the dropfilters is applied to the first layer of a CNN, the generalization error becomes higher than the CNN with no dropfilters in any convolutional layers. So, we did not apply dropfilters to the first convolutional layer of our model. It results in 2.62% higher accuracy than a model applied dropfilters in all convolutional layers. Then, we applied the dropfilters to a CNN based on the inception module to recognize handwritten Hangul. The overview of the dropfilters applied to the inception module is illustrated in Fig.2.

We will examine related works in Section 2, and Section 3 describes the dropout applied to convolutional layers. Section 4 presents the dropfilters and we describe experiments about HHR with the CNN based on the inception module in Section 5. And finally, we will conclude in Section 6.

## II. Related work

CNN has been applied successfully to various recognition tasks. However, there are few relevant offline HHR studies that use CNN-based models [6] [7]. The models used in the two studies is the basic LeNet structure [15] with two modified objective functions. In the former study, classification based on Mean Squared Error(MSE) is performed and, an additional weight parameter $\alpha$ is introduced to reduce the imbalance between positive and negative signals generated during the backpropagation process. In the latter study, they proposed a discriminative function to enhance discriminative power of similar characters. Then, hybrid learning is performed by combining the discriminative function with cross entropy loss as an objective function.

Recently, various architectural models of the CNN have been explored. The notable famous models are the GoogLeNet [8] and the Residual Networks [16]. Each has unique architectural components. GoogLeNet uses the inception module for two purposes. The first purpose is to decrease of the computational cost by using dimensional reduction masks. The second purpose is to enhance representation power by multi-scaled filter masks in an inception module. Specifically, GoogLeNet achieves about 10 percent high top-5 recognition accuracy for the Imagenet dataset with 12 times fewer parameters than the architecture of [3]. Residual network uses residual connections for stacking many hidden layers. Also, Inception v4 [17] which is a combination of the inception module and the residual connection produces the-state-of-the-art accuracy on various classification tasks. Therefore, it is plausible to use the CNN model based on the inception module for handwriting recognition which requires minimal computation overheads.

As the model structure becomes deeper and the number of parameters increases, the dropout, which is a typical regularization technique, was proposed to avoid the overfitting problem [10]. The dropout technique is a method to remove some neurons of hidden layers with an arbitrary probability $p$ when learning the model. This reduces the number of model parameters involved in the actual learning. As a result, the overfitting problem is alleviated. Another important interpretation is that the ensemble learning effect of a randomly selected sub-network can be applied, which is very helpful for improving the recognition accuracy. The DropConnect can be regarded as a generalized concept of dropout [18]. From the study, The weights of fully connected layers are dropped with an arbitrary probability $p$. Then the model is learned with the subsets of weights. The dropout and the dropconnect are mainly applied to fully connected layers.

Regularization techniques for the convolutional layer have also been extensively studied. First, there is the stochastic pooling method that performs a pooling operation randomly by changing the existing deterministic pooling to stochastically [19]. Also, the maxout network, which selects maximum activations as the output of the layer, is also applied to the convolutional layer [20]. Then, combined with dropdout technique, the maxout shows to reduce generalization error. In addition, Gal and Ghahramani [21] propose the Bayesian convolutional neural network that learns the CNN using Monte Carlo-dropout technique. They formulate a dropout network as an approximation of a bayesian network. Further, Thompson et al. [13] proposed the spatial dropout technique which drops the several entire channels of feature map of convolutional layers with probability $p$ instead of dropping each neuron.

## III. Dropout in convolutional layers

The dropout technique in the fully connected layer has contributed greatly to reduce the generalization error of the model. Convolutional layers differ from fully connected layers in their structure and operation. For example, in the convolutional layer, a feature map consists of 3-dimensional tensor and weight parameters called filter masks are shared to each

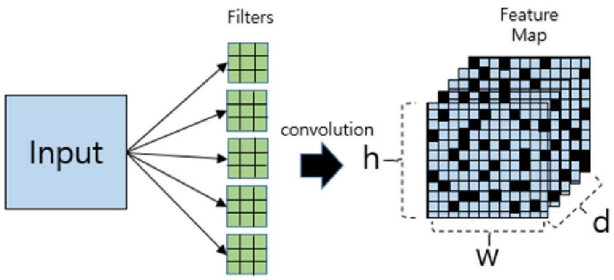


Fig. 3. Diagram depicting dropout applied to each node in a feature map of a convolutional layer. The black shaded squares mean the pixels are set to zero by dropout.

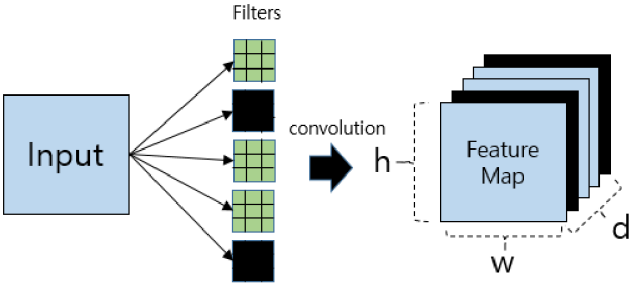


Fig. 4. Diagram depicting dropfilters applied to convolutional layer. The black shaded square on the feature map indicates the dropped channel.

pixel. Therefore, we need to consider about the proper way to use dropout in convolutional layers. The basic approach is illustrated in Fig.3 where, $w$ and $h$ represent the width and height of the output feature map after convolution with mask filters, respectively, and $d$ represents the depth of the output feature map. In the feature map, the black shaded squares mean the pixels are set to zero by dropout. This is a method of dropping arbitrary pixels on a feature map that pass through convolutional layers. Consider a feature map of a convolutional layer. Let, $\mathbf{r} \in \{0,1\}^{W \times H \times D}$ denote the vector of $r$ sampled from the Bernoulli distribution with probability $p$, $\mathbf{y} \in \mathbb{R}^{W \times H \times D}$ denote the output feature map. The $W$, $H$, and $D$ are the width, height, and depth of the output feature map. $z$ is the input feature for convolutions, and $\mathcal{M}$ is mask filters. $d \in \{1, ..., D\}$ means the number of mask filters which is same as the depth of the output feature map after convolutional operations. $\mathbf{b}$ is biases. Now, we can calculate the $i$, $j$, $d$'th value of output feature map by following this:

$$r_{ijd} \sim Bernoulli(p),$$
$$y_{ijd} = r_{ijd} \times (b_d + \sum_{w=1}^{W^{'}} \sum_{h=1}^{H^{'}} \sum_{k=1}^{K} z_{(i+w)(j+h)k} \times \mathcal{M}_{whkd}) \quad (1)$$

Where $W^{'}$, $H^{'}$, and $K$ are the width, height, and depth of mask filters. At the (1), we considered that the width and height of input feature map are same with the output feature map by using zero padding.

## IV. Dropfilters in convolutional layers

Tompson et al. claimed that applying standard dropout(where each pixel of convolutional feature maps is dropped out independently) did not prevent the overfitting [13]. They assumed that natural images exhibit strong spatial correlation, and the activations of a feature map are also strongly correlated. Therefore, they proposed a method called spatial dropout that drops out all adjacent pixels instead of dropping one pixel at a time. In the research, they considered the adjacent pixels as a whole channel of a feature map, so they dropped each channel of feature map instead of each pixel. We further see the spatial dropout in different point of view of paying attention to the ensemble learning effect of sub-networks. In other words, since the core parameters of a CNN are convolutional masks, it is more intuitively plausible that sub-networks of a CNN could be represented by different combinations of convolutional masks. Therefore, we dropped the arbitrary mask filters with probability $p$. This is defined as following.

$$r_d^{'} \sim Bernoulli(p),$$
$$y_{ijd} = b_d + \sum_{w=1}^{W^{'}} \sum_{h=1}^{H^{'}} \sum_{k=1}^{K} z_{(i+w)(j+h)k} \times \mathcal{M}_{whkd} \times r_d^{'} \quad (2)$$

Where $\mathbf{r}^{'} \in \{0,1\}^d$ is dropfilters parameter sampled from the Bernoulli distribution with probability $p$. The $d \in \{1, ..., D\}$ is the $d$'th value of dropfilters parameter $\mathbf{r}^{'}$. Since, the dropfilters sets arbitrary channels to zero values, the feature map becomes sparse like a hidden layer with dropout. So, the model with dropfilters could be robust to noises and reduce generalization error. The result of (2) is same with spatial dropout which drops arbitrary channels, but we consider spatial dropout technique for convolutional layers as the ensemble learning effect, which is a different interpretation with the spatial correlation. The overview of this method is illustrated in Fig. 4.

## V. Experiments

### A. Effect of the Dropfilters in Each Layer

Before we apply the dropfilter to a CNN based on the inception module, we need to check effects of the dropfilters in each layer. This is motivated by a study about an understanding of convolutional neural networks [22]. Zeiler and Fergus [22] revealed that features of each convolutional layer have meaningful properties for understanding why the CNNs recognize well. From their research, lower layers of a CNN react to basic patterns of input such as lines or curves. On the other hand, upper layers react very class-specific patterns such as the face of a person of the wheel of a car. Considering the facts revealed by [19], we expected that applying the dropfilters to lower layers of CNNs is not necessary to reduce generalization error. So, we made an assumption as following.

TABLE I
STRUCTURE OF A CNN BASED ON THE INCEPTION MODULE

| Layers | Output | 1x1 | 3x3 reduce | 3x3 | 5x5 reduce | 5x5 | Pool proj | # of parameters |
|---|---|---|---|---|---|---|---|---|
| Incept1 | 30x30x112 | 1x32 | 1x48+48 | 9x48x32 | 1x16+16 | 25x16x16 | 1x32 | 20,416 |
| Incept2 | 15x15x240 | 1x112x64 | 1x112x64+64 | 9x64x64 | 1x112x16+16 | 25x16x48 | 1x112x64 | 79,440 |
| Incept3 | 8x8x448 | 1x240x64 | 1x240x64+64 | 9x64x128 | 1x240x32+32 | 25x32x128 | 1x240x128 | 245,343 |
| Incept4 | 1x1x512 | 1x448x128 | 1x448x96+96 | 9x96x128 | 1x448x48+48 | 25x48x128 | 1x448x128 | 443,536 |
| FC1 | 512*384+384 | - | - | - | - | - | - | 196,992 |
| FC2 | 384*520+520 | - | - | - | - | - | - | 200,200 |
| Total | - | - | - | - | - | - | - | 1,185,927 |

TABLE II
RESULTS OF EXPERIMENTS WHEN THE DROPFILTERS APPLIED TO EACH LAYERS RESPECTIVELY

| Model | Accuracy | | |
|---|---|---|---|
| | Training | Validation | Test |
| No drop | 1.0 | 0.8182 | 0.8061 |
| Conv1 | 0.9998 | 0.7984 | 0.7907 |
| Conv2 | 0.9998 | 0.8221 | 0.8189 |
| Conv3 | 0.99987 | 0.8397 | 0.8364 |
| Conv4 | 0.9996 | 0.8273 | 0.82 |

- Mask filters of lower layers of a CNN model consists of basic features such as a straight line or curved line, which is not a redundant or overfitting factor.

As a method of verifying our assumption, we have applied the dropfilters technique from the first layer to the last layer of the convolutional layers, respectively, to find the hidden layer contributing a substantial regularization effect. We used the CIFAR10 dataset which consists of 10 classes and 60,000 natural images in total [14] for the experiment. The models used for the verification experiments are composed of four convolutional layers each followed by max-pooling and two fully connected layers. All convolutional layers have 128 filters with a stride of 1 pixel, and the size of each filter is all 3×3. This is because we wanted to check exact effects of the dropfilters applied each convolutional layer. Therefore, we fixed all other variables for accurate experimentation. The two fully connected layers have 512 and 10 units respectively, and the second fully connected layer is the softmax layer. The drop rate of all dropfilters is set to 0.5. Also, the activation function of all layers is the ReLU. For verification, we learned the model for 200 epochs, and the result is illustrated in Table II. In the Table II, No drop means the model without the dropfilters, and Conv1∼4 mean the models which are applied the dropfilters to each convolutional layer. Experimental results show that the dropfilters increased the generalization error when applied only to the first layer. On the other hand, when the dropfilters applied individually to the second, third and fourth layer, the generalization error was reduced. Further, we performed an experiment to measure the contribution of the dropfilters applied concurrently to several layers. For the experiment, we set a model which has 64, 128, 256 and 512 mask filters in each convolutional layer, and other settings are same with the model which was used to the verification test. The result is illustrated in Table III. At the Table III, All drop means the dropfilters applied to all convolutional layers, and Drop 234 means the dropfilters applied simultaneously to 2∼4 convolutional layers. From the two experiments of this subsection, we could verify our assumption. Therefore, in all subsequent experiments, the dropfilters was not applied to the first layer.

TABLE III
RESULTS OF EXPERIMENTS WHEN THE DROPFILTERS APPLIED TO LAYERS CONCURRENTLY.

| Model | Accuracy | | |
|---|---|---|---|
| | Training | Validation | Test |
| No drop | 0.9998 | 0.8125 | 0.8052 |
| All drop | 0.998 | 0.8467 | 0.8351 |
| Drop 234 | 0.9992 | 0.8671 | 0.8613 |

### *B. The Inception Module with the Dropfilters*

The inception module is a core component of GoogLeNet, which has been widely used until recently because of its good recognition accuracy and reduced computational cost. The inception module has various representational power by applying multi-scaled mask filters and concatenating it as illustrated in Fig2. As depicted in Fig2, the inception module has several stages of convolutional operation in one module block, so it is also a matter to consider how to apply the dropfilter technique to the inception module. We decided not to apply the dropfilter to the depth reduction performed before 3×3 and 5×5 convolutions. This is because, in the case of depth reduction, it plays a role in reducing the depth of the feature map, so there maybe significant information losses when the several reduction masks are removed. Therefore, we did not apply the dropfilters technique to the reduction masks.

### *C. Data Description*

We collected handwritten Hangul data on our own from a wide range of demographic (e.g. ages, jobs, and gender). The examples of collected data are illustrated in Fig1. As depicted in Fig1, in the case of handwritten Hangul, there are various characters which have simple strokes to complex strokes. Also, many characters appear similar to each other. So, it can be seen that among the many kinds of handwriting, the handwritten Hangul is difficult to recognize with high accuracy. The data consists of 520 classes of 500 characters

TABLE IV
STRUCTURE OF A CNN BASED ON THE LENET

| Layers | Size of filters / stride | # of parameters |
|---|---|---|
| Conv1 | 5x5x64+64 / 1 | 1664 |
| Pool1 | 2x2 / 2 | - |
| Conv2 | 5x5x64x128+128 / 1 | 204,948 |
| Pool2 | 2x2 / 2 | - |
| Conv3 | 4x4x128x256+256 / 1 | 524,544 |
| Pool3 | 2x2 / 2 | - |
| Conv4 | 4x4x256x512+512 / 1 | 2,097,664 |
| pool4 | 2x2 / 1 | - |
| FC1 | 512x384+384 | 196,992 |
| FC2 | 384x520+520 | 200,200 |
| Total | - | 3,226,012 |

TABLE V
EXPERIMENTAL RESULTS FOR HANDWRITTEN HANGUL RECOGNITION

| Model | Accuracy % | | |
|---|---|---|---|
| | Training (a) | Test (b) | (a) - (b) |
| LeNet NDF | 99.847 | 92.573 | 7.274 |
| LeNet DF | 97.943 | 94.673 | 3.270 |
| Inception NDF | 99.876 | 94.762 | 5.114 |
| Inception DF (proposed) | 97.737 | 95.848 | 1.889 |

per class and 260,000 characters in total. This is the same as the character classes belonging to SERI95a in [6]. For the experiment, 400 characters per class were selected for the training set, and 100 characters per class were selected for the test set. Therefore, the total number of training sets and test sets was 208,000 and 52,000 characters, respectively.

### *D. Handwritten Hangul Recognition*

Through our previous experiments, we examine effective method of applying dropfilters in the convolutional layers. Based on this, we extensively validate recognition experiments on handwritten Hangul data set. In our experiments we did not apply any data augmentation techniques. Ablation experiments were performed on LeNet-based CNN with and without the dropfilters. Then, we also performed experiments for the CNN based on the inception module with and without the dropfilters. Table I and Table IV illustrates the structure of CNNs based on the inception module and the LeNet, respectively. To verify that a CNN based on the inception module can achieve higher accuracy with fewer parameters, the model was constructed with about 3 times fewer parameters than the CNN based on LeNet. The dropfilters was applied to all convolutional layers except for the first convolutional layer in the two models. Drop rates were all set to 0.5. In both models, the activation function was the same as ReLU, and the batch normalization technique [23] was applied to the convolutional layer of both models for rapid learning. In the FC1 layer of both models, the dropout technique is applied with probability of 0.5, and the FC2 layer was composed of the softmax layer. The loss function was cross-entropy. The result is depicted in Fig. 5, where NDF means that the dropfilters was not applied to the model. Conversely, DF means the model applied the dropfilters. Tr

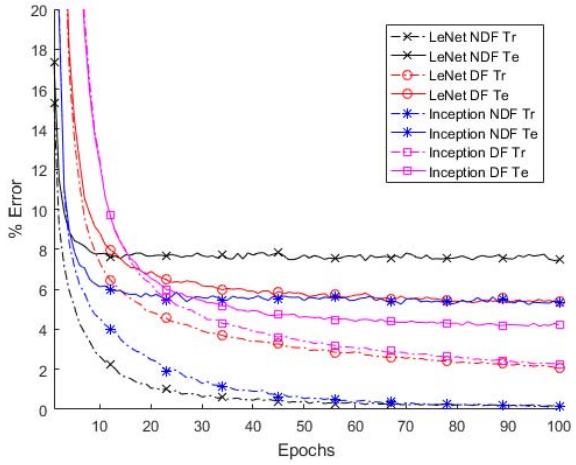


Fig. 5. Results for Handwritten Hangul Recognition. Dash-dot lines represent training errors, and solid lines are test errors. Also, the models are distinguished by line markers and colors

and Te represent training and test, respectively. Also, LeNet and Inception mean models shown in Table IV and Table I, respectively. From the Table V, we could verify effects of both the dropfilters and the inception module. (a) - (b) in Table V means the difference between training and test accuracy, which could be called as the degree of overfitting. At the experiment, models based on the inception modules achieve higher recognition accuracy than models that are composed of the LeNet structure. Surprisingly, the LeNet with dropfilters achieves test accuracy of 94.673, which is comparable to the test accuracy of a model based on inception modules. However, the model based on the inception modules achieves not only the highest recognition accuracy of 95.848% but also the lowest degree of overfitting of 1.889% with about only 30% of parameters relative to the LeNet.

## VI. CONCLUSION

In this paper, we address the offline HHR task with a model based on inception modules with a novel sparsifying method named dropfilters to cope with resource-constrained environments(e.g. smartphones, tablet PCs, etc.). The proposed model obtained 3.275% higher recognition performance while the number of parameters is approximately three times fewer than that of the LeNet, a classical sparse convolutional network. Also, since the model reduces generalization error well, it could be applied to data written by various kinds of people with a smaller amount of data.

However, when the dropfilter is applied to other very deep models such as residual net, we observed that the model could not be trained. It is because there were too many combinations of sub-networks to be trained. Solving this problem is considered as a future work. Also, we are working on multi-language(e.g. Japanese, Chinese, etc.) recognition including special symbols(e.g. exclamation mark, question mark, star,

heart, text-based emotion expression character such as Emoji, etc.) with this single model for practical deployment.

## ACKNOWLEDGMENT

This work was partly supported by the ICT R&D program of MSIP/IITP. [2017-0-00162, Development of Human-care Robot Technology for Aging Society] and [2015-0-00310-SWStarLab, Autonomous Cognitive Agent Software That Learns Real-Life with Wearable Sensors]. This work was also partly supported by Samsung Electronics, Co., Ltd..

## REFERENCES

[1] S.-J. Cho and J. H. Kim, "Bayesian network modeling of hangul characters for online handwriting recognition," in *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*. IEEE, 2003, pp. 207–211.

[2] C. Bahlmann, B. Haasdonk, and H. Burkhardt, "Online handwriting recognition with support vector machines-a kernel approach," in *Frontiers in handwriting recognition, 2002. proceedings. eighth international workshop on*. IEEE, 2002, pp. 49–54.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[4] P. Sermanet, S. Chintala, and Y. LeCun, "Convolutional neural networks applied to house numbers digit classification," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 3288–3291.

[5] A. Yuan, G. Bai, L. Jiao, and Y. Liu, "Offline handwritten english character recognition based on convolutional neural network," in *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*. IEEE, 2012, pp. 125–129.

[6] I.-J. Kim and X. Xie, "Handwritten hangul recognition using deep convolutional neural networks," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 18, no. 1, pp. 1–13, 2015.

[7] I.-J. Kim, C. Choi, and S.-H. Lee, "Improving discrimination ability of convolutional neural networks by hybrid learning," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 19, no. 1, pp. 1–9, 2016.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[9] W.-Y. Kang, B.-H. Kim, and B.-T. Zhang, "Hangul handwriting recognition using deeper convolutional neural networks based on inception modules," *Korea Computer Congress 2016 (KCC2016)*, pp. 883–885, 2016.

[10] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[11] A. Ng, "Sparse autoencoder," *CS294A Lecture notes*, vol. 72, no. 2011, pp. 1–19, 2011.

[12] H. Wu and X. Gu, "Towards dropout training for convolutional neural networks," *Neural Networks*, vol. 71, pp. 1–10, 2015.

[13] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 648–656.

[14] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

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

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[17] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," *arXiv preprint arXiv:1602.07261*, 2016.

[18] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 1058–1066.

[19] M. D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," *arXiv preprint arXiv:1301.3557*, 2013.

[20] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. C. Courville, and Y. Bengio, "Maxout networks." *ICML (3)*, vol. 28, pp. 1319–1327, 2013.

[21] Y. Gal and Z. Ghahramani, "Bayesian convolutional neural networks with bernoulli approximate variational inference," *arXiv preprint arXiv:1506.02158*, 2015.

[22] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.