Pattern Recognition Letters 131 (2020) 172–177

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

High cursive traditional Asian character recognition using integrated adaptive constraints in ensemble of DenseNet and Inception models

Amin Jalali, Minho Lee*

Kyungpook National University, 80 Daehak-ro, Buk-gu, Daegu 41566, Republic of Korea

ARTICLE INFO

Article history:
Received 30 June 2019
Revised 28 December 2019
Accepted 8 January 2020
Available online 9 January 2020

Keywords:
Deep learning
Adaptive regularizer
Korean document
Pattern recognition

ABSTRACT

In this paper, we propose integrated adaptive sensitivity and robustness terms for the cost function of a convolutional neural network (CNN). The sensitivity term considers the slight variations and high frequency components of the input image samples. It distinguishes between images that look similar but belong to different classes. This regularizer is designed to enhance the between-class distance which is a biological definition for the simple cells of the visual system. On the other hand, the robustness term is used to develop a more stable CNN structure against disturbances and perturbations. The robust term provides better within-class features because it recognizes images that look different but are actually from the same class. The robust term symbolizes the complex cell characteristics of the visual system. The coefficients of the sensitivity and robustness regularization terms are adaptively tuned along with the network parameters using gradient descent. Two optimizers are assigned to tune the parameters: one for tuning the model parameters and the other one to adjust the sensitivity and robustness coefficients. This approach is applied to Korean traditional documents for character classification. The results show better within- and between-class classification ability for highly complex character styles with imbalanced number of samples.

1. Introduction

The Chinese language has historically been a part of the traditional Korean language. There is a need to analyze the impact of traditional Chinese scripts on the Korean language throughout all of the changes that have occurred to the Korean language over time to form the modern Korean language. The research in this area will provide more information about the culture heritage and increase the historical and linguistic background, which will assist in interpreting handwritten historical scripts. There are many recognition difficulties when attempting to translate the traditional Korean writing (Hanja) into the modern Korean language (Hangul). Variant handwriting styles, size variability, the high similarity between characters, the large number of characters, and the existence of compound characters in Chinese script make it challenging for researchers to explore this issue and achieve better recognition. To conduct research on traditional Korean scripts, the academy of Korean studies has been collecting image samples, which are public accessible.[1],[2] All of these historical documents are handwritten manuscripts and have difficulties in recognition. The documents have image degradation including document aging and issues with the quality of the ink such as ink dispersion due to the passage of time. Moreover, the dataset is highly imbalanced. There are just a few samples for some characters, while for others there are larger numbers of samples. Moreover, in some cases, the documents have a low-quality because they date back to several hundred years ago. Because of the passage of time, the ink has been dispersed all over the edges, which makes them difficult to read. These highly degraded samples lower the recognition performance. In this research, we perform offline handwritten character recognition in which the scripts are captured optically by a camera. With the increase in the utilization of convolutional neural networks (CNN) in different fields, researchers have begun using it for character recognition. GoogleNet was utilized along with the directional feature maps, Gabor filters, and histogram of gradients which assisted in improving the recognition of Chinese characters [15]. Another study employed a hierarchical CNN model for the recognition of similar confusable Chinese characters [12]. The

* Corresponding author.
E-mail address: mholee@knu.ac.kr (M. Lee).

[1] http://kostma.aks.ac.kr/segment/segmentList.aspx.
[2] Dataset download link: https://github.com/Amin-AI/Regularizations-DNN.

https://doi.org/10.1016/j.patrec.2020.01.013
0167-8655/© 2020 Elsevier B.V. All rights reserved.

model had two parts, a deep network block and hierarchical block. The former was used to distinguish the inter-class characters, and the latter was used for the intra-class classification of the characters. The fine exploration of similar characters was performed in the hierarchical block by multiple parallel CNN classifiers, which looked for nuances and critical areas to distinguish characters.

Taking into account these improvements in the CNN topology, computational time, and accuracy, the traditional Korean scripts written in Chinese characters are highly imbalanced. To the best of our knowledge, this is the first dataset that has been provided by the Academy of Korean Studies for research on the translation of traditional Korean archives (Hanja) into the modern Korean language (Hangul). Moreover, the previous datasets provided in the literature have fewer complexities. More than half of the data have few samples because they are segmented directly from documents coming from a variety of sources. The samples include all sorts of disturbances, including ink dispersion, extensive cursive styles, low-quality resolution, and complex backgrounds. A deep powerful CNN structures such as ResNet [1] and Inception [11] are essential to address these complexities in the dataset. In addition, regularization methods should be investigated to enhance the recognition performance. Therefore, to achieve better character recognition, we present regularization terms to be adaptively employed in the cost function of the training algorithm to diminish the impact of input degradation over the output error to ensure robust recognition. The proposed approach simultaneously and adaptively uses both sensitivity and robustness regularizations. Because there are many similar confusable characters and there are various types of disturbances within these characters in real old documents, we incorporate the sensitivity in the cost function of the CNN to try to distinguish these characters from each other. These confusable characters have high similarities and they visually look similar because of added noise and cursive style transformations, but they actually belong to different character classes. The other concern in this research is characters with less similar characteristics. These characters seem to belong to two different classes but in fact, they are from the same class. We attempt to incorporate robustness in the cost function of the training algorithm to deal with the recognition of characters with lower similarities that belong to the same class. In order to leverage the advantages of both the sensitivity and robustness terms, it is necessary to adaptively find appropriate coefficients for each of them in the training process. We defined the initial values of the coefficients and then trained them through the learning process of the network. In order to train the model parameters, one adaptive gradient descent optimization is used to tune the parameters of the CNN, and another is utilized to adjust the coefficients of the sensitivity and robustness terms. The optimization methods are the same, but the learning ratios and gradient values are different from one another. The parameters of the model (i.e., weights and biases) and coefficients of the regulators (sensitivity and robustness coefficients) have different scales. Thus, we need two optimizers, each defined with specific values. Once each network is trained with the various regularization terms, we use ensemble learning to obtain the maximum confidence score out of all the trained models to obtain higher accuracy.

A literature review of the CNN models applied in character recognition is provided in Section 2. The proposed adaptive sensitivity and robustness regularizations approach is presented in Section 3. We present the experimental results in Section 4, and finally we conclude in Section 5.

2. Literature review on application of various CNN topologies in Chinese character recognition

A study analyzed misclassified Chinese characters using top-1-votes, which represented the votes for the character class given by most humans [6]. In their studies they drew the conclusion that some samples were written in a way that could be recognized as belonging to multiple classes and could be considered to be multi-label samples. These samples had medium top-1-votes, and their confidence scores caused their characteristics to be regarded as multi-label samples because of their confusing labels. Moreover, the samples with low top-1-votes were either written in an immense cursive writing style or wrongly written, which made it difficult for them to be properly recognized. Another study [2] found that characters with low recognition accuracy have similar shapes, features, and commonalities. A tiny difference in the stroke length, shape, or direction in characters with the same radicals can cause confusion during recognition. A method called DropSample [13] was introduced for samples with low confidence-scores. At each iteration of the training process, the samples with low confidence-scores are more likely chosen at training process in the next iteration. This approach efficiently involves the samples with more ambiguity and low confidence scores in the training process. This approach was the inspiration of Leitner's learning box, in which materials that require a greater learning efforts appear in the learning process more often than those that require less efforts. The integration of shape normalization and direction-decomposed feature maps with CNN structures was also presented [14]. The direction-decomposed features were good representations because the characters were made of basic directional strokes. The shape normalization method diminishes the within-class variance but incurs a reverse impact of stroke direction distortion. Therefore, they utilized a new adaptation layer to gradually decrease the mismatch between training and test data. The adaptation was performed by adding a layer to the CNN structure. This layer, which was called style transfer mapping (STM), adapted new styles of handwritten characters to the model with a small number of samples of that particular writing style in an unsupervised way. In order to apply the adaptation layer, each group of data with the same consistent handwriting style was fed into the same unsupervised adaptation layer to calculate the probability distribution. In our traditional Korean scripts, the characters are from various style sources and are all mixed together. The writer-specific data styles are not classified separately. Thus, it is not possible to employ the adaptation layer in our work to attain a performance enhancement.

3. Proposed integrated adaptive sensitivity and robustness regularizations in cost function

The primary version of robustness regularization in the cost function of a training algorithm has been shown to improve the performance of the image recognition [4]. To make a robust error cost function, the average value of the derivative of the activation function of the neurons of the last layer are added to the cost function of the error back-propagation learning algorithm. This term causes data samples in the decision boundaries to be given greater weights by adding the derivatives of the activation function outputs and creates better features, which prevents a drastic update of the network weights and allows general exploitation. Moreover, the primal sensitivity constraint was introduced [5], which incorporates a sensitivity term in the cost function of a CNN to emphasize the slight variations and high frequency components in highly blurred input image samples. The proposed cost function has a sensitivity part which is the reverse of the average of the derivatives of the activations, and subsequently the total error is minimized by the gradient descent method during the learning process. Because of the proposed sensitivity term, the data samples at the decision boundaries appear more often on the middle band or high gradient part of the activation function.

The sensitivity term may distinguish samples with high similarities. We attempted to increase the between-classes distance

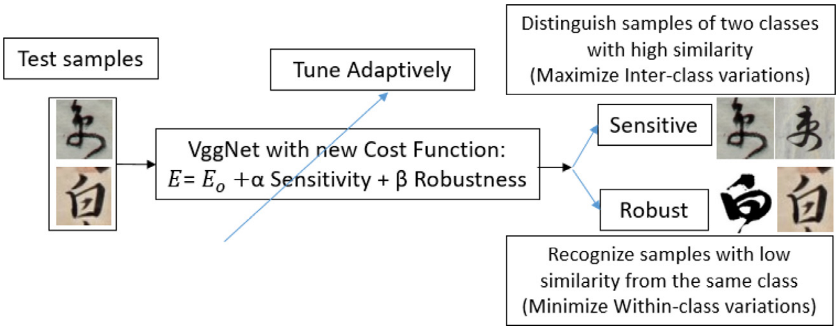
Test samples
Tune Adaptively
Distinguish samples of two classes with high similarity (Maximize Inter-class variations)
VggNet with new Cost Function:
$E = E_o + \alpha$ Sensitivity + β Robustness
Sensitive
Robust
Recognize samples with low similarity from the same class (Minimize Within-class variations)

Fig. 1. Integrated adaptive sensitivity and robustness regularizations in cost function of CNN structure

of samples using the sensitivity regularizer. The sensitivity operation was inspired by the simple cells of the human visual system. Highly similar samples that belong to two different classes are usually misclassified as being in the same class. The baseline models might not differentiate these similar samples because they have very analogous features and small intra-class differences. These small features and high frequency components may come into play when using the sensitivity term, leading to better separation. On the other hand, the robustness regularizer term has the ability to recognize images that look different but actually belong to the same class. Biologically, the complex cells of the visual system extract more sophisticated features to come up with this part of recognition. The robustness term may provide lower within-class distance and tune the weights accordingly. To obtain generalized robust classification, the impact of the outliers has to be reduced to ensure valid classification even though the input values are changed by noise and disturbances [4]. In the proposed approach the average of the derivatives of the activations is employed for the output error as an added term with a relative significance factor, and the total error is minimized by the error back propagation learning. There are two parts of the weight updates, i.e., the back propagated error and gradient of the hidden neuron penalty term. Data samples in the decision boundaries are given greater weights by adding the additional term. This follows the Hebbian learning rule. The Hebbian term is also multiplied by the derivative of the activation function, which avoids a drastic update of the network weights. The theoretical justification was presented in literature [10]. Moreover, the sensitivity term highlights the slight changes in the input images and attempts to consider the small differences in the samples such as strokes and curvatures. This could be used to distinguish the samples of classes that look alike. In the proposed approach, by considering these two regularization terms, i.e., the sensitivity and robustness terms, we are able to accommodate both penalty terms in one cost function alongside the output error rate. This method could change the gradients for the better if the coefficient of each penalty term is properly set. Adjusting the coefficients heuristically is a cumbersome task. These coefficients, i.e., α and β, correspond to the relative significance impacts of the sensitivity and robustness terms, respectively, as shown in Fig. 1. Because the weights and biases are trained using gradient descent, α and β could also be trained by the gradient descent error back-propagation approach. We defined two separate training algorithms that used the same approach but different settings. The parameters of the training algorithm could be learning rate, momentum and weight decay rate. One back-propagation training algorithm is designed for the weights and biases and the other is designed for training the coefficients of the regularizers. Training the coefficients adaptively can assist in automatically finding the optimum values.

In Fig. 1, the model is first trained with the new cost function, including the integrated adaptive sensitivity and robustness terms. Then, in the test stage, the test samples are inserted into the trained model for evaluation. There are two rows of images on the right side of Fig. 1. In the first row, the images are from different classes, but because they look similar, the models usually recognize them as the same class. However, by incorporating the sensitivity term in the cost function of the training algorithm, we aim to distinguish these samples with high similarities that belong to two different classes. On the other hand, in the second row, there are images that might not look alike but belong to the same class. We intend to recognize these two samples with lower similarities and classify them as the same class using the robustness term in the cost function of the training algorithm. These regularizations are performed automatically because they are defined in the cost function, which is tuned during the training phase.

The weight update rule in error back-propagation is performed by Eq. (1), in which $w_{np}^{(l)}$ is the weight between the p^{th} neuron of layer $l-1$ and the n^{th} neuron of layer l. The other parameter is η_1, which represents the learning rate.

$$\left(w_{np}^{(l)}\right)_{new} = \left(w_{np}^{(l)}\right)_{old} + \eta_1 \nabla w_{np}^{(l)} \tag{1}$$

$\nabla w_{np}^{(l)}$ is the derivative of the weights for layer l. It is calculated by the partial derivative of the output error with regard to the weights in Eq. (2). E_o is the output error. $\delta_n^{(l)}$ denotes the error value propagated from the last layer to l^{th} layer. $h_p^{(l-1)}$ is the input feature map of the l^{th} layer.

$$\nabla w_{np}^{(l)} = -\frac{\partial E_o}{\partial w_{np}^{(l)}} = -\delta_n^{(l)} h_p^{(l-1)} \tag{2}$$

Eq. (3) represents the average value of the derivatives of the activations of the last layer of the CNN denoted by R as the regularizer. In order to acquire the sensitivity term, 1 is divided by R. In Eq. (4), m is the margin value to prevent the term from having an infinitive value. Thus, the sensitivity penalty is $\tilde{E_S}$.

$$R = \frac{1}{N}\sum_{i=0}^{N}\left(f'\left(\sum_j w_{ij}^{(l)} h_j^{(l-1)}\right)\right) \tag{3}$$

$$\tilde{E_S} = \frac{1}{R+m} \tag{4}$$

The changes in the weights are found in Eq. (5) by using the sensitivity penalty term in the cost function.

$$\nabla w_{np}^{(l)} = -\frac{\partial \tilde{E_S}}{\partial w_{np}^{(l)}} = -\left(\delta_n^{(l)} + \eta \frac{R'}{R^2}\right) h_p^{(l-1)} \tag{5}$$

Based on Eq. (6), to obtain the robustness penalty term, the regularizer term R is added to conventional loss value E_o. The robustness penalty term is denoted by $\tilde{E_R}$.

$$\tilde{E_R} = E_o + R \tag{6}$$

By incorporating the robustness term in the cost function, the effect of the changes in the weights is found in Eq. (7).

$$\nabla w_{np}^{(l)} = -\frac{\partial \tilde{E_R}}{\partial w_{np}^{(l)}} = -\left(\delta_n^{(l)} + \eta R h_n^{(l)}\right) h_p^{(l-1)} \tag{7}$$

The proposed approach uses the sensitivity and robustness terms integrated with the conventional loss value. Eq. (8) denotes the total error $\tilde{E_T}$, in which α and β determine the relative significances of the sensitivity and robustness terms, respectively. Thus, we extend the loss function of the CNN with the incorporation of Lambda (λ) as shown in Eq. (9) to train our model. λ represents the significance coefficient.

$$\tilde{E_T} = E_o + \alpha \tilde{E_S} + \beta \tilde{E_R} \tag{8}$$

$$\tilde{E_T} = \lambda E_o + (1 - \lambda)\left(\alpha \tilde{E_S} + \beta \tilde{E_R}\right) \tag{9}$$

Based on Eq. (1), the proposed gradients calculation of the weights and biases is presented in Eq. (10). $(\delta_n^{(l)})_T$ denotes the error value propagated from the last layer to l^{th} layer using new loss value $\tilde{E_T}$.

$$\nabla w_{np}^{(l)} = -\frac{\partial \tilde{E_T}}{\partial w_{np}^{(l)}} = -\left(\delta_n^{(l)}\right)_T h_p^{(l-1)} \tag{10}$$

The update rule for α and β are performed by Eqs. (11) and (12), in which $(.)_{new}$ is the updated value for each coefficient in current iteration and $(.)_{old}$ is the coefficient value from the previous iteration. $\nabla\alpha$ and $\nabla\beta$ are the gradients of the coefficients. These are calculated by the partial derivative of the total error ($\tilde{E_T}$) with regard to the coefficients values. The gradient calculation of these coefficients (i.e., $\nabla\alpha$ and $\nabla\beta$) are based on Eq. (10). The learning rate to update these coefficients (η_2) is different from the learning rate of the weights and biases (η_1).

$$(\alpha)_{new} = (\alpha)_{old} + \eta_2 \nabla\alpha \tag{11}$$

$$(\beta)_{new} = (\beta)_{old} + \eta_2 \nabla\beta \tag{12}$$

We utilize the ensemble approach to obtain higher performance using the integration of sensitivity and robustness regularization terms in the cost function of the training algorithm. Each of these regularization terms could modify model features according to the characteristics of that particular regularizer. Hence, several models are trained using the introduced regularization terms to optimize the final performance. Four ensemble methods are considered in our experiments which include ensemble of models with different: (1) augmentation methods (2) activation functions (3) regularizations (4) sensitivity and robustness settings shown in Table 4. Each model trained separately and the weights are stored. In the test stage, the confidence scores are obtained passing each test image through each model and the Softmax function. The confidence scores are the output probabilities generated by Softmax function. By considering the number of classes as n, there is a matrix of size ($models \times n$) at the output of the Softmax function. The maximum value on each column of this matrix is selected to obtain the highest confidence score for each class among all the trained models. Then, the output matrix after the Maximum value block is ($1 \times n$). Finally, to find the output class of the proposed approach, the argument of the maximum value of ($1 \times n$) matrix is returned.

4. Experimental results & discussion

The dataset has 4844 classes, with a significant imbalance between the numbers of samples in these classes, which ranges from one sample to several hundred samples. The training split includes 341,300 segmented samples and the test split includes 1825 samples extracted from 300 documents. The dataset has classes in which the samples have high intra-class distances. There are many characters in the dataset that have several styles of writing. The recognition of the classes with various styles causes recognition ambiguity because of their similarities to other characters. Furthermore, the small number of samples in a class with various styles adds additional complexity and confusion in the recognition task. This dataset is highly imbalanced because it is directly segmented from traditional Korean manuscripts written in the old Chinese language.

Table 1
Augmentation methods and their parameters.

Augmenters	Parameters
Cropping	percent=(−0.1, 0.2)
Elastic transform	alpha=(0.3, 1.15), sigma=0.25
Affine transform	scale=”x”: (0.9, 1.2), ”y”: (0.9, 1.2)
Translation percent	”x”: (−0.1, 0.15), ”y”: (−0.15, 0.1)
Rotation	(−15, 15)
Shearing	(−5, 5)
Sharpening	alpha=(0, 1), lightness=(0.85, 1.2)
Adding to Hue	(−15, 15)
Contrast norm	(1.2, 2.0), per channel=0.5
Piecewise affine	scale=(0.02, 0.05)
Perspective	scale=(0.01, 0.1)

We use random augmentations to compensate for the lack of insufficient training samples. A sequence of augmentation steps is defined in the augmentation process and some of these are randomly applied to the images. The augmenters include cropping, elastic transformation, affine transformation, translation, rotation, shearing, sharpening, changing the hue and saturation of images, contrast normalization, piecewise affine transformation, and perspective transformation. As listed in Table 1, the cropping operation crops the images by −10% to 20% of their height/width (Minus sign in −10% represents the left side of the image and plus sign in 20% denotes the right side of the image). Elastic transformation moves each pixel individually based on distortion fields and generates shape variations. The term ”sigma” defines the smoothness of the distortion field, and ”alpha” is its strength. Affine transformation scales images to 90% to 120% of their size individually per axis. Translation percent moves the pixels by −10% to 15% in the x-direction and −15% to 10% in the y-direction. The rotation operation rotates by −15° to 15°. Shearing by −5° to 5° is applied. The sharpening kernel runs over each image with lightness in the range of (0.85, 1.2) and mixes the result with the original image using the alpha range. In the next operation, ”Adding to Hue and Saturation”, a value in the range of −15° to 15° is added to each pixel in the HSV space. Contrast normalization changes the contrast of the images by moving pixel values away from or closer to 128. The direction and strength is in the range of (1.2, 2.0) and applied to 50% of the channels randomly. Piecewise affine transformation moves parts of the image around on a scale of (0.02, 0.05). Perspective transformation applies a random four-point perspective transform to the image. Each point has a random distance from the image corner, derived from a normal distribution with sigma on the scale of (0.01, 0.1). The dataset is already noisy, so we just augment the classes whose samples are less than fifteen samples. If the number of samples in any class is less than fifteen and greater than seven, we apply a random mixture of the augmentation techniques in Table 1 to augment each samples five times. If the number of samples are less than seven images, each sample is augmented to ten more images. Moreover, we performed the discrete cosine transform (DCT) and principal component analysis (PCA) to project a random image in each class onto DCT and PCA subspace for data augmentation [8]. DCT maps the features matrix into a smaller uncorrelated directions while keeping the global Euclidean structure.

Table 2
Performance of VggNet, ResNet, DenseNet, and InceptionNet with different augmentations, activation functions, and regularization methods.

Operation	Methods	Vgg Net	Res Net	Dense Net	Incp. Net
Augment	Baseline	65.7	75.5	75.7	77.1
	Seq. of Aug.	66.8	76.3	76.4	77.7
	PCA/DCT	66.2	75.8	75.9	77.2
	Fusion	67.1	76.4	76.7	77.8
Activation function	SReLU	66.9	76.1	75.9	77.6
	PReLU	67.1	76.2	76.5	77.5
	ELU	66.9	76.4	76.8	77.3
	Swish	67.3	76.6	76.4	77.8
	ReLU	67.1	76.4	76.7	77.8
Regularize	Sensitivity	68.3	77.3	77.6	78.7
	Robustness	68.1	77.2	77.2	78.6
	Sen+Rob	68.8	77.6	77.8	**79.4**

Table 3
Effect of sensitivity and robustness coefficients on performance.

Hyper-parameters	test1	test2	test3
Sensitivity Coefficient (α)	9	25	30
Robustness Coefficient (β)	17	25	9
Lambda (λ)	0.7	0.4	0.5
Learning Rates of Coefficients (η_2)	0.03	0.06	0.09
Accuracy	79.4	79.7	**79.9**

PCA also generates the orthogonal projection vectors while maximizing the variance of the projected vectors.

Another issue involves the confusable similar characters. The only difference in these pairs is usually a stroke or small additional parts. Because the dataset contains handwritten characters written in various styles, the skew and cursive styles make it even more complicated to achieve good recognition. This dataset contains samples with occlusions and many characters with incomplete parts in low-quality images. The differences between these pairs could be small changes in the character width or length, subtle changes in the stroke direction, nuance differences in the local structure and writing style, or similar confusing radicals. The main point of this paper is to use the integrated regularization methods to tune the structure in a way to give better recognition performance for such characters. The well-recognized characters are among the characters where the stroke-structure is well-shaped and are not usually among the confusable characters. The small differences in the stroke lengths and orientations in confusable similar character pairs make it difficult for them to be well-recognized. The issues that might lead to misrecognition of the test samples are noise, disturbance, a heavy cursive handwritten style, low resolution of the samples, skewness, and illegible writing.

Table 2 lists the results of a comparison of the different models that we used for traditional Korean script recognition. We employed four baseline models such as VggNet [9], ResNet50 [1], DenseNet169 [3], and Inception-V3 [11]. The baselines' parameters are the same as the referenced papers. The first row shows different augmentation methods such as sequence of augmenters (Table 1), PCA/DCT [8], and fusion of both. The second row demonstrates the model's performance by using different activation functions [7]. The third row presents the regularization constraints such as sensitivity, robustness, and the integration of sensitivity and robustness. The results in the first row show that using the fusion of both sequence of augmenters and PCA/DCT results in better feature representation and higher performance. We utilized the model with fusion augmentation in the first row to investigate the effect of activation functions in the second row. The experiments indicate that there is not an activation function that outperforms the others in all models. The third row presents the effect of applying regularization methods. Comparing the the regularizers show that the incorporation of the sensitivity and robustness together (sen+rob) in the cost function will lead to better feature extraction and further classification. Moreover, the Inception model outperforms the others with highest accuracy. The experiments are performed with dropout ratio of 30%. The batch size is 64 samples. The models are trained for 60 epochs. The learning rate (η_1) is 0.001, and the learning rate of the coefficients (η_2) is listed in Table 3. A network pre-trained on the ImageNet dataset is utilized. The parameters for all the models are the same to make a fair comparison. The hyper-parameters' values are the extra added parameters for the proposed model, which have the values listed in Table 3. These hyper-parameters are from Eq. 9. As demonstrated by the accuracy values listed in Table 3, the initial values of the sensitivity and robustness coefficients could have an influence on the output performance. The hyper-parameters, namely the sensitivity coefficient (α), robustness coefficient (β), Lambda value (λ), and learning rates of these coefficients (η_2), should be initialized at the beginning of the learning process. The coefficients are trained by back-propagation through the training process to the optimum values that minimize the cost function. As listed in Table 3, three testbeds were designed. Each column presents the values of each of these hyper-parameters, and the last row shows the performance of the VggNet that includes the sensitivity and robustness regularization in the cost function.

Table 4 presents the performance of ensemble models for the Inception network which is the best performing baseline. The first row is the ensemble of three different augmentation methods in the first row of Table 2, namely, sequence of augmenters, PCA/DCT, and fusion. The second row is the ensemble of five Inception networks with different activation functions in the second row of Table 2, namely, SReLU, PReLU, ELU, Swish, ReLU. The third row is the ensemble of different regularization methods in the third row of Table 2, namely, sensitivity, robustness, and integration of both. The fourth row is the ensemble of three versions of Inception networks trained with integrated sensitivity-robustness with different settings. The specifications of each of the three models used in this method are denoted in Table 3. Table 4 shows that ensemble approach that uses the integration of sensitivity and robustness at the same time results in the highest performance among the proposed methods.

Fig. 2 shows the properly recognized samples by integrated sensitivity-robustness in (a) Inception network and (b) ResNet network. The first row shows the test samples with their corresponding Unicodes (true labels). The second row shows the target images that are well-recognized by proposed method. The last row shows the misrecognized outputs by Vanilla model. For example, in the second column in (a), the test sample with the Unicode of 62*DC* is very similar to the sample with the Unicode of 6367. Our goal is to distinguish these two samples and classify them into two different classes. Plain Inception network cannot properly distinguish test sample 62*CD* but classifies it as 6367. In other words, Inception network places both of them in one class. However, by utilizing the proposed method, the accurate class is effectively determined, and different classes are separated from one another. By investigating the third column in (b), we can infer that the proposed

Table 4
Performance of ensemble approaches for Inception network.

Ensemble methods with Inception model	Acc
3 models with different augmentation methods	78.6
5 models with different activation functions	79.5
3 models with different regularizations	81.4
3 models with different sen-rob settings	**82.1**

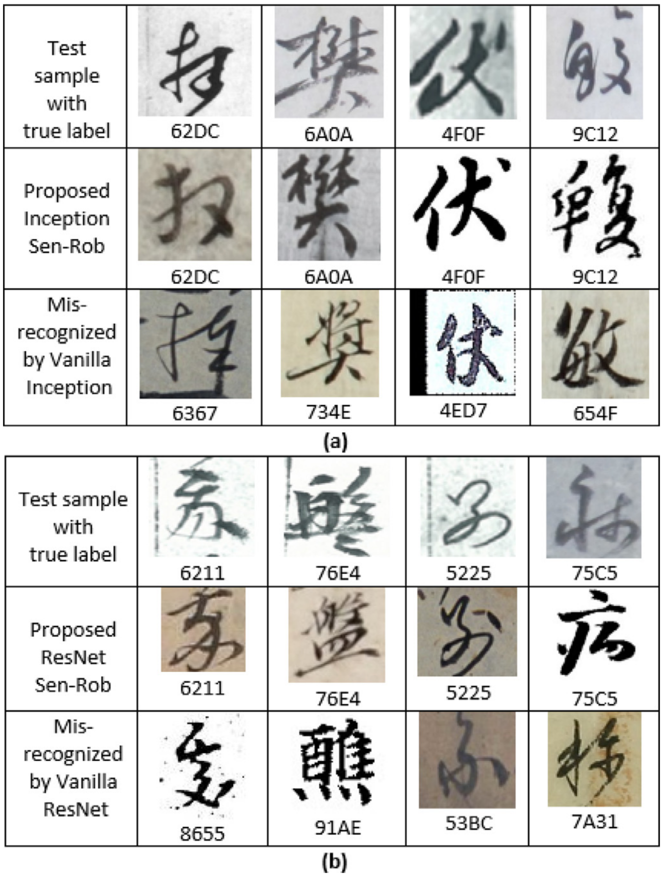
Test sample with true label
62DC
6A0A
4F0F
9C12
Proposed Inception Sen-Rob
62DC
6A0A
4F0F
9C12
Mis-recognized by Vanilla Inception
6367
734E
4ED7
654F
(a)
Test sample with true label
6211
76E4
5225
75C5
Proposed ResNet Sen-Rob
6211
76E4
5225
75C5
Mis-recognized by Vanilla ResNet
8655
91AE
53BC
7A31
(b)

Fig. 2. Well-recognized samples by using integrated sensitivity-robustness regularizations in (a) Inception and (b) ResNet structures

model can properly recognize the more complicated samples that have rather high intra-class distance variance and low inter-class distance variances. These samples in the third column share many commonalities, but should be distinguished and assigned to other classes. The ensemble of all the models that utilize the regularization penalty terms produces a superior output. Our model attempts to decrease the intra-class distance and increase the inter-class distance for better classification. For example, considering the second column in (b), we realize that the test sample with Unicode 6211 not only has low similarity to its target in the second row but also has high similarity to samples of another class with Unicode 8655. In this case, the utilization of both sensitivity and robustness result in better recognition.

5. Conclusion

The methodology presented in this study was an ensemble of adaptive sensitivity and robustness regularizations in the cost function of the Inception structure. The sensitivity regularizer was incorporated in the cost function to distinguish between highly similar images that in fact belong to different classes. Such highly similar images share many commonalities. In order to find their small nuances (i.e., strokes, dots, and curvatures), the sensitivity regularization was employed. Moreover, there are images of a class that have low similarities. These are usually misrecognized and assigned to different classes, but they in fact belong to the same class. We attempt to recognize them as one entity using the robustness term. The proposed method integrated these two regularizers with their own relative coefficients and employed them along with the output error of the model in the cost function. Further, the coefficients of the sensitivity and robustness terms were adaptively tuned with the model parameters. This provided appropriate values to allow each penalty term to be in harmony as one whole unit structure. Finally, several models tuned by regularizers were combined as an ensemble to obtain superior recognition performance on traditional Korean scripts written in the old Chinese language. Therefore, the proposed ensemble adaptive sensitivity and robustness regularizations promote better performance in deep structures.

Declaration of Competing Interest

None.

Acknowledgments

This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2017M3C1B6071400)(50%), by the Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (2016-0-00564, Development of Intelligent Interaction Technology Based on Context Awareness and Human Intention Understanding)(50%)

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patrec.2020.01.013.

References

[1] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[2] M. He, S. Zhang, H. Mao, L. Jin, Recognition confidence analysis of handwritten chinese character with cnn, in: Document Analysis and Recognition (ICDAR), 2015 13th International Conference on, IEEE, 2015, pp. 61–65.
[3] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
[4] A. Jalali, G. Jang, J.-S. Kang, M. Lee, Convolutional neural networks considering robustness improvement and its application to face recognition, in: International Conference on Neural Information Processing, Springer, 2015, pp. 240–245.
[5] A. Jalali, R. Mallipeddi, M. Lee, Sensitive deep convolutional neural network for face recognition at large standoffs with small dataset, Expert Syst. Appl. (2017).
[6] K. Liang, L. Jin, Z. Xie, X. Xiao, W. Huang, A comprehensive analysis of misclassified handwritten chinese character samples by incorporating human recognition, in: Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on, 1, IEEE, 2017, pp. 460–465.
[7] G. Maguolo, L. Nanni, S. Ghidoni, Ensemble of convolutional neural networks trained with different activation functions, arXiv:1905.02473 (2019).
[8] L. Nanni, S. Brahnam, S. Ghidoni, G. Maguolo, General purpose (genp) bioimage ensemble of handcrafted and learned features with data augmentation, arXiv:1904.08084 (2019).
[9] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556 (2014).
[10] L. Soo-Young, J. Dong-Gyu, Merging back-propagation and hebbian learning rules for robust classifications., Neural Netw. 9 (7) (1996) 1213–1222.
[11] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.
[12] Q. Wang, Y. Lu, Similar handwritten chinese character recognition using hierarchical cnn model, in: Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on, 1, IEEE, 2017, pp. 603–608.
[13] W. Yang, L. Jin, D. Tao, Z. Xie, Z. Feng, Dropsample: a new training method to enhance deep convolutional neural networks for large-scale unconstrained handwritten chinese character recognition, Pattern Recognit. 58 (2016) 190–203.
[14] X.-Y. Zhang, Y. Bengio, C.-L. Liu, Online and offline handwritten chinese character recognition: a comprehensive study and new benchmark, Pattern Recognit. 61 (2017) 348–360.
[15] Z. Zhong, L. Jin, Z. Xie, High performance offline handwritten chinese character recognition using googlenet and directional feature maps, in: Document Analysis and Recognition (ICDAR), 2015 13th International Conference on, IEEE, 2015, pp. 846–850.