Discovering Class-wise Trends of Max-pooling in Subspace

Yuchen Zheng, Brian Kenji Iwana and Seiichi Uchida
Department of Advanced Information Technology, Kyushu University, Fukuoka, Japan
Email: {yuchen, brian}@human.ait.kyushu-u.ac.jp, uchida@ait.kyushu-u.ac.jp

***Abstract*—The traditional max-pooling operation in Convolutional Neural Networks (CNNs) only obtains the maximal value from a pooling window. However, it discards the information about the precise position of the maximal value. In this paper, we extract the location of the maximal value in a pooling window and transform it into "displacement feature". We analyze and discover the class-wise trend of the displacement features in many ways. The experimental results and discussion demonstrate that the displacement features have beneficial behaviors for solving the problems in max-pooling.**

***Index Terms*—Convolutional neural networks; Max-pooling; Displacement feature.**

I. Introduction

The max-pooling operation is a common step in modern deep convolutional neural networks (CNNs), which is often introduced to obtain translation-invariant representations. Many models that utilized the max-pooling operation in their architectures have been proposed [1], [2], [3], [4], [5], which demonstrates some good properties of the max-pooling operation in many ways. Typically, the introduction of max-pooling layers in CNNs has helped to satisfy some properties by allowing a network to be somewhat spatially invariant to the position of features [6]. However, the traditional max-pooling operation only preserves the maximal value in a pooling window and discards the location of the corresponding maximal value, which looses the spatial information of the original data.

In this paper, we extract the maximal values and their corresponding locations simultaneously from the pooling windows. In the condition of multiple maximal values in a pooling window, we average all of the location coordinates as the final location information of the maximal value and call it a "displacement feature". Based on the displacement features, we mine the class-wise behaviors and trends. Firstly, we visualize the displacement features based on a HSV color model and a t-SNE [7] visualization. Next, we summarize the displacement features on all feature maps in each class, and illustrate the cumulative histograms to understand the distribution of the displacement features. Then, we introduce Principal Component Analysis (PCA) to further analyze the properties of the displacement features and compare the similarity of class-wise subspaces spanned by the first N largest principal components. Finally, we compute the confusion matrices that are obtained by the recognition only using the displacement features. The discussion of the displacement features demonstrates many desirable properties and bring some merits for handwritten digit recognition task. The contribution of this paper are summarized as follows.

- We extract the maximal values and their locations from the pooling windows simultaneously.
- This paper analyzes the properties of the displacement features from each category in many ways.
- This paper mines the class-wise trend of max-pooling in the subspaces.

The rest of the paper is organised as follows: Sect. II discusses some work related to our own, we introduce the displacement features in Sect. III, and finally give the analysis and discussion in Sect. IV, while Sect. V concludes this paper with remarks and future work.

II. Related Work

In recent years, many models have been proposed based on the max-pooling operation for handwritten recognition task [8], [9], [10], [11], [12]. Among these works, most of them use the CNNs with max-pooling layers in their models. In [8], Ciresan et al. proposed a handwriting recognition model based on relaxation convolutional neural network (R-CNN) and alternately trained relaxation convolutional neural network (ATR-CNN) with two max-pooling layers. This work achieved good performance on recognizing offline handwritten Chinese characters. Due to the powerful learning ability of deep CNNs, Zhong et al. proposed a framework by using AlexNet [2] and GoogleNet [3] with directional feature maps for handwritten recognition task, which also contained many max-pooling layers to build the networks. In the ICDAR 2013 Chinese handwriting recognition competition [13], 10 groups submitted 27 systems for five tasks: extracted features, online/offline isolated character recognition, online/offline handwritten text recognition. Among these submitted systems, Deep CNNs with several max-pooling layers based methods have shown superiority in both isolated character recognition and handwritten text recognition [14]. However, these works only utilized the max-pooling operation to construct the systems without discussion and modification on the max-pooling operation.

To overcome the defects of the max-pooling operation, many researchers focus on modifying the pooling layers in many ways. Zeiler et al. proposed a simple and effective method which replaces the conventional deterministic max-pooling operations with a stochastic procedure, randomly picking the activation within each pooling region according to a multinomial distribution, given by the activities within

the pooling region [15]. This approach is hyper-parameter free and achieve good performance without data augmentation. Malinowski and Fritz proposed a flexible parameterization that allows for a richer set of possible pooling regions and extended the learnable pooling regions to the events recognition task with object banks as high level features [16]. Murray and Perronnin proposed a novel pooling mechanism that involves equalizing the similarity between each patch and the pooled representation, called Generalized Max Pooling (GMP), which can provide significant performance gains with respect to heuristic alternatives such as power normalization [17]. Qian et al. proposed max-pooling positions (MPPs) as an effective discriminative feature for traffic sign recognition [18]. In this work, For a 2×2 pooling window, they defined a quaternary encoding, '1000, 0100, 0010, 0001', where the '1' represents the position of maximal value in pooling windows.

In addition, there are also many works analyze and discuss the max-pooling operation in many ways [19], [20], [21], [22], [23]. However, these works are not based on the location information of the maximal values in the pooling windows. In our paper, we analyze and discuss the displacement features in many ways and discover the class-wise trends of the max-pooling operation in subspaces.

III. Displacement Features from Max-Pooling Operation

In this section, we introduce how to extract the displacement features from pooling windows. Then, we present how to address the problem if there are more than one maximal value in a pooling window.

A. Extracting Displacement Features from Convolutional Layers

In traditional CNNs, a max-pooling layer often appears after a convolutional layer. For instance, in Fig. 1, for a 4×4 convolutional feature with 2×2 pooling size, there are 4 pooling windows that are represented by 4 colors. The bold numbers are the maximal values in pooling windows. Then, we obtained a 2×2 pooling feature from a 4×4 convolutional feature. However, traditional max-pooling operation only obtained the maximal value from a pooling window and does not record where the maximal value is, which may lose the spatial information of the original features. In order to preserve these crucial information, we record the location of the maximal value from a pooling window and transform it into a "displacement feature". The detail of this procedure is presented in Fig. 2. Compared with the even case, the odd case has the central unit "(0,0)". Then, we divide the displacement features into two parts, "dis-x" and "dis-y" shown in Fig. 3, which represents the horizontal and vertical directions of the displacement features respectively.

B. Multiple Maximal Values Condition

The previous section only discusses the case that only single maximal value in a pooling window. However, in some cases, a pooling window may contain more than one maximal value. Fig. 4 presents some cases with multiple maximal values conditions. To address this problem, we first record all locations of the maximal values in a pooling window. Then, we calculate the mean value of all locations as the final displacement feature. If all units are maximal values in a pooling window, through simple computation, the final displacement feature is "(0,0)".

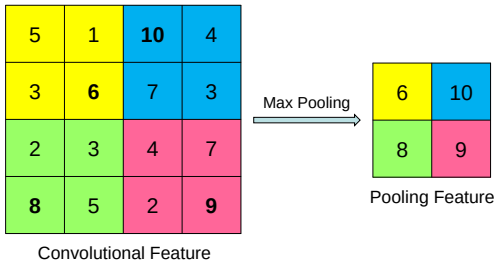

Fig. 1. An example of max-pooling with kernel size 2×2 stride 2.

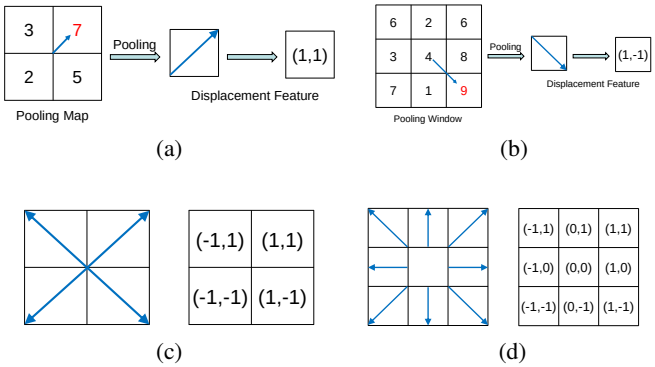

Fig. 2. (a) Even case: extracting the displacement features from the pooling window, where the pooling size is 2×2. (b) Odd case: extracting the displacement features from the pooling window, where the pooling size is 3×3. (c) Displacement features in even case. (d) Displacement features in odd case.

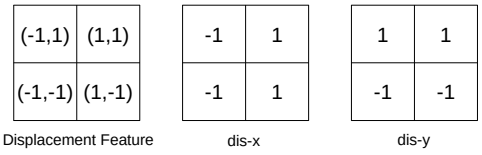

Fig. 3. Example of extracting dis-x and dis-y from the displacement feature.

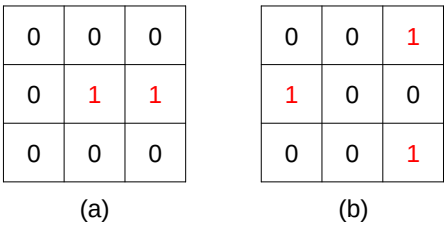

Fig. 4. Different cases of multiple maximal values conditions. (a) Displacement features: (0,0),(1,0). Mean displacement feature: $(\frac{1}{2},0)$. (b) Displacement features: (1,1),(-1,0),(1,-1). Mean displacement feature: $(\frac{1}{3},0)$.

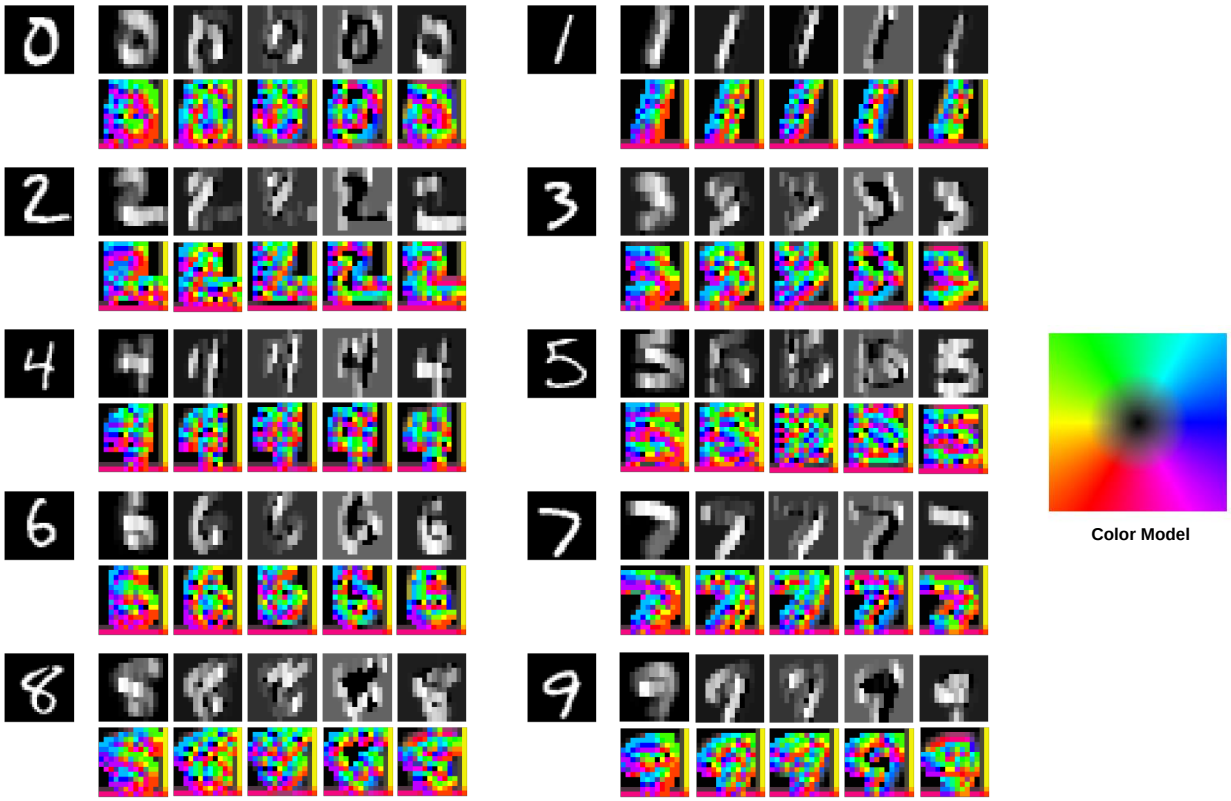

Fig. 5. Visualization of the pooling features and the displacement features on different classes. Here, the pooling size is 5×5, the first row represents the original image and corresponding pooling features, the second row represents the displacement features, each column represents one convolutional filter. The visualization of displacement features is based on the right color model whose color and intensity denote direction and average length of displacement feature.

IV. Experiment and Discussion

In this section, we discuss and discovery the class-wise trend of the displacement features in MNIST dataset [1]. At first, We extracted the displacement features from the first convolutional layer in a normally pretrained CNN. The architecture of the normal CNN contains two convolutional (the kernel sizes are 5×5 with stride 1) and pooling layers (the kernel sizes are 5×5 with stride 2), two fully connected layers (1024 hidden units), and the final layer is a softmax layer. The cross entropy was used as the loss function, and we minimized it by Adam with a 1×10^{-4} learning rate, the batch size and epoch are set to 50 and 20,000, respectively. Next, we visualized the displacement features by a color model and the t-SNE model. Then, we illustrated the cumulative histograms of the displacement features on all feature maps and observe the distribution of the displacement features both in same and different classes. Next, we used PCA method and computed the similarity of class-wise subspaces spanned by the first N principal components. Finally, we utilized the displacement features for recognition task and computed the confusion matrix to compare with the normal CNN.

[1]http://yann.lecun.com/exdb/mnist/

A. Visualization of Displacement Features

In order to discover the class-wise trend of the displacement features, we visualized the displacement features based on a HSV color model. The visualization results are shown in Fig. 5 and 6. In Fig. 5, we can see that each pooling feature represents one kind of local feature that is extracted by different convolutional filters. The corresponding displacement features record the location information that describes the position of the maximum. In addition, the displacement features on the different classes have large dissimilarities. The shapes and boundaries are very similar to the corresponding pooling features and original images, which may help for recognition task in some cases. From Fig. 6, we can see that the samples in the same class often have the similar behaviors. For instance, in the first filter of all class "0" samples, the top left corners are in blue direction, the top right corners are in green direction, the bottom left and right corners are in purple and red directions, respectively. The samples in different classes have much different behaviors. It is easy to distinguish them by only using the displacement features.

In Fig. 7 and 8, we also visualized the displacement features of the test dataset on 2D space by t-SNE [7]. We can see that there is a very clear class-wise trend of the displacement features. The distance between the intraclass samples are smaller than the interclass samples. In Fig. 7, the class "1",

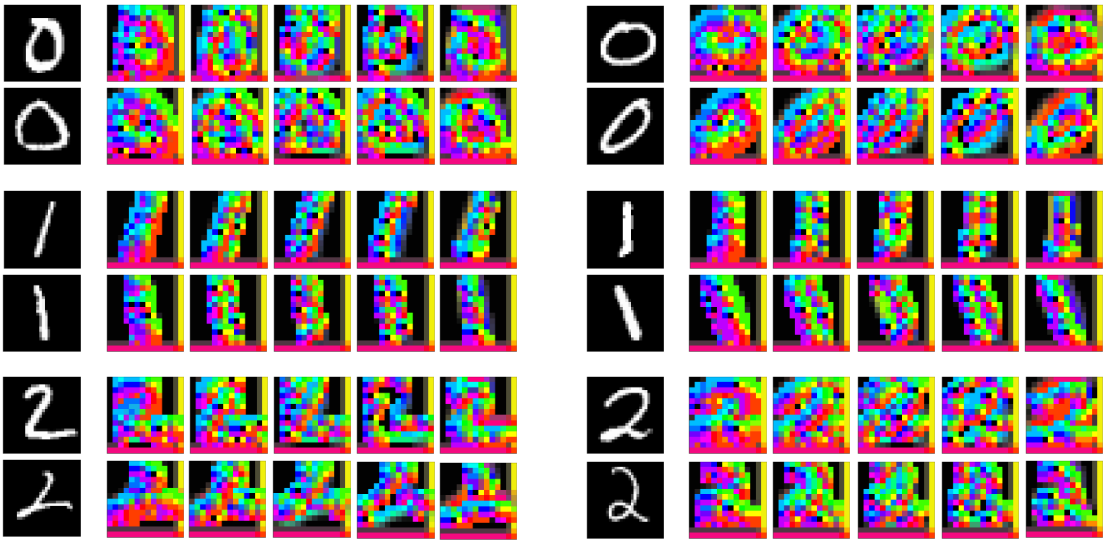

Fig. 6. Visualization of the displacement features on the different samples that are in the same class.

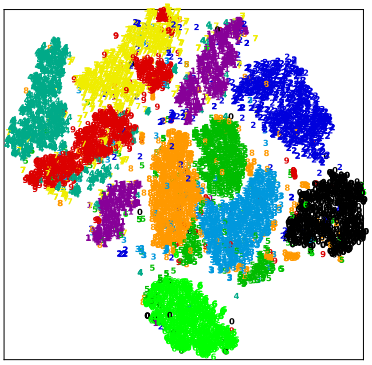

Fig. 7. Visualization of the displacement features (dis-x) in horizontal direction by t-SNE.

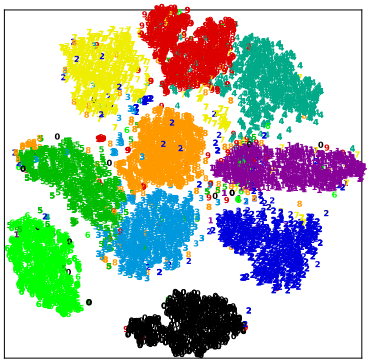

Fig. 8. Visualization of the displacement features (dis-y) in vertical direction by t-SNE.

"4", "5", "9" have more clusters than other classes. But, this phenomenon does not appear in Fig. 8. Therefore, it may mean that the vertical displacement features are better than the horizontal displacement features for classification tasks.

B. The Distribution of Displacement Features

In order to further observe the behaviors of the displacement features, we illustrated the cumulative histograms of the displacement features, which accounts the coordinate points of the displacement features of different samples on all feature maps (filters). To obtain more information of the displacement features, the pooling size of the first pooling layer is set to 5×5 with stride 2. The distribution of the displacement features in the first pooling layer is shown in Fig. 9. Due to the space limitation, we just illustrated the 12 samples from 3 classes.

In Fig. 9, Due to the pooling size being 5×5, the displacement features "dis-x" and "dis-y" belong to $[-2, 2]$. We can see that the samples in the same class have the similar distributions. For the class "0", there are some values around the central point (0,0) and their quantities are about 100. For the class "1", there is a relatively horizontal line on all samples. It is very similar to class "1", because the displacement information is rare in vertical direction. For the class "2", the distribution is more scattered than the class "0" and "1". In addition, the most value in all samples is (0,0), that is because the most features in original images are backgrounds.

C. Class-wise Similarity of Displacement Features in the PCA Subspaces

We also measured the displacement features in PCA subspaces. At first, we trained different PCA models on the samples that are in the same classes. Then, we preserved 10% of eigenvalues for each PCA model to build the PCA subspaces. Based on these subspaces, we calculated the class-wise similarity matrix for each feature map. The similarity can be defined by using the canonical angles,

$$\cos\theta_i = \sup_{\substack{\mathbf{x}_i \perp \mathbf{x}_j,\, \mathbf{y}_i \perp \mathbf{y}_j \\ 1 \le i,\, j \le p}} \frac{\mathbf{x}_i^T \mathbf{y}_i}{\|\mathbf{x}_i\| \|\mathbf{y}_i\|} \tag{1}$$

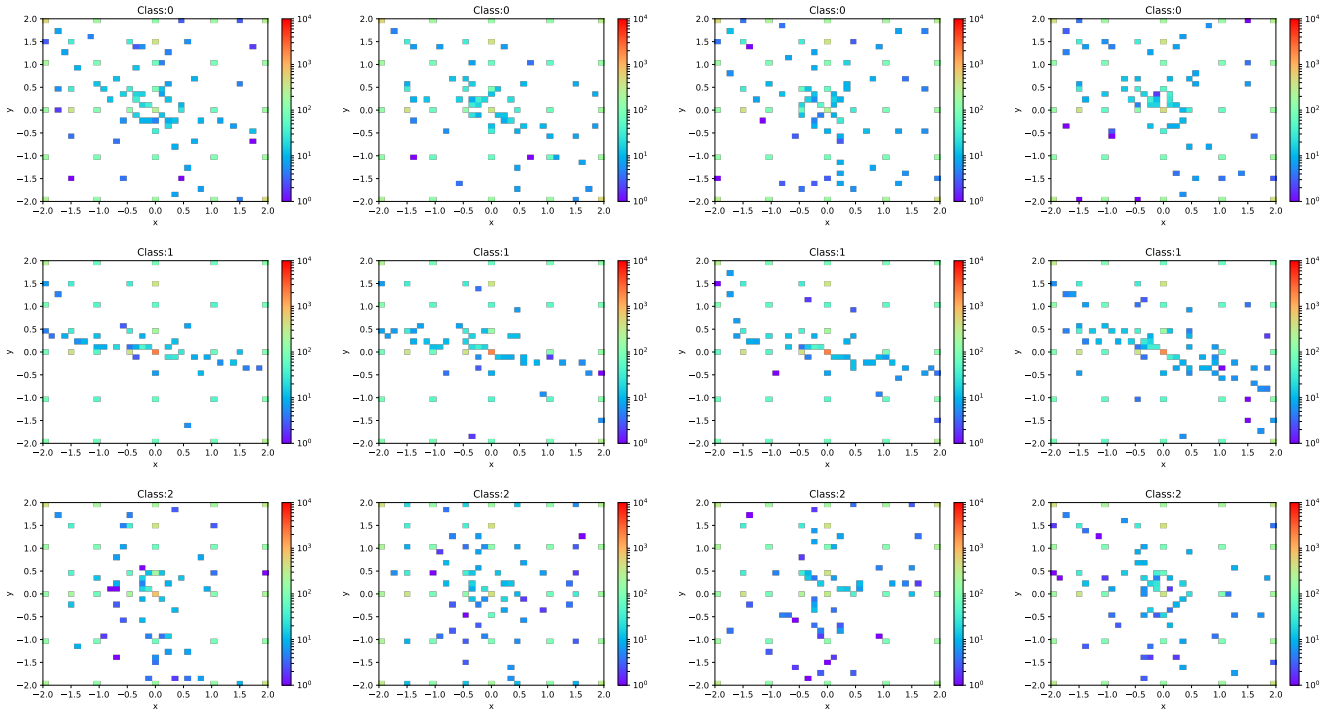

Fig. 9. The distribution of displacement features from class "0", "1" and "2" (each row represents 4 different samples in the same class). The color histograms represent the number of the displacement features on corresponding coordinate points.

Here, $\mathbf{x} \in \mathbf{P}, \mathbf{y} \in \mathbf{Q}$, $\mathbf{P}$ and $\mathbf{Q}$ are two PCA subspaces, $\mathbf{P}, \mathbf{Q} \in \mathbb{R}^n$, dim $\mathbf{P} = p \leq$ dim $\mathbf{Q} = q$. Then the similarity can be defined as,

$$S = \frac{1}{p} \sum_{i=1}^{p} \cos \theta_i^2 \tag{2}$$

If two PCA subspaces completely coincide with each other, all canonical angles will be 0 and S equals to 1. The similarity gets smaller as the two spaces separate. Finally, the similarity is zero when the two subspaces are orthogonal to each other [24]. The similarity matrix are shown in Table. I and II. Due to the space limitation, we just presented the similarity matrix of two convolutional filters. We can see that the similarities in different classes are relatively small, which means that the displacement features are discriminative in different classes. In addition, the similarity of class"4" and "9", "7" and "9" are larger than others, but they are far away from 1.

D. Classification by Using Displacement Features

Finally, we used the displacement features (dis-x and dis-y) for classification task on MNIST dataset. We also used the previous network that discards the first convolutional and pooling layer (because the displacement features are the output of the first pooling layer and have the same size of the first pooling layer features). Then, we compared the confusion matrix that only using the pooling features and only using the displacement features. The confusion matrix are shown in Table. III, IV and V. We can see that, only using the displacement features for classification also can obtain the relatively high accuracies. In class "5", "6" and "8", the horizontal displacement features are better than the pooling features for classification, and in class "2" and "5", the vertical displacement features are better than the pooling features. In other cases, the pooling features perform better than the displacement features. Therefore, we can reasonably speculate that combining the displacement features and pooling features may improve the performance of some specific tasks.

V. Conclusion

In this paper, we extract the displacement features that record the location information of the maximal values in pooling windows. Then, we discover the class-wise trend of the displacement features in many ways. Through the analysis and discussion, the displacement features may improve the performance of some specific tasks. For the future work, we plan to adopt some other techniques to enhance the displacement features and combine the displacement features with pooling features in some ways.

Acknowledgement

This research was partially supported by MEXT-Japan (Grant No.J17H06100) and NTT Communication Science Laboratories.

References

[1] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column Deep Neural Networks for Image Classification," in *CVPR*, 2012.
[2] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *NIPS*, 2012.

TABLE I
SIMILARITY MATRIX ON THE FIRST CONVOLUTIONAL FILTER IN PCA SUBSPACES.

C	0	1	2	3	4	5	6	7	8	9
0	1	0	0.11	0.10	0.04	0.08	0.21	0	0.15	0.05
1	0	1	0	0.05	0.01	0.01	0.01	0.14	0	0.03
2	0.11	0	1	0.14	0.04	0.01	0.01	0.03	0.09	0.04
3	0.10	0.05	0.14	1	0.01	0.05	0.01	0.04	0.14	0.06
4	0.04	0.01	0.04	0.01	1	0.03	0	0.19	0.16	0.38
5	0.08	0.01	0.01	0.05	0.03	1	0.03	0.02	0.15	0.06
6	0.21	0.01	0.01	0.01	0	0.03	1	0	0.01	0.01
7	0	0.14	0.03	0.04	0.19	0.02	0	1	0.05	0.33
8	0.15	0	0.09	0.14	0.16	0.15	0.01	0.05	1	0.09
9	0.05	0.03	0.04	0.06	0.38	0.06	0.01	0.33	0.09	1

TABLE II
SIMILARITY MATRIX ON THE SECOND CONVOLUTIONAL FILTER IN PCA SUBSPACES.

C	0	1	2	3	4	5	6	7	8	9
0	1	0.12	0.12	0.16	0.04	0.11	0.12	0.04	0.14	0.08
1	0.12	1	0.11	0.08	0.12	0.08	0.07	0.24	0.29	0.08
2	0.12	0.11	1	0.10	0.04	0.05	0.03	0.07	0.12	0.01
3	0.16	0.08	0.10	1	0.05	0.18	0.01	0.04	0.14	0.09
4	0.04	0.12	0.04	0.05	1	0.04	0.10	0.07	0.08	0.21
5	0.11	0.08	0.05	0.18	0.04	1	0.04	0.03	0.18	0
6	0.12	0.07	0.03	0.01	0.10	0.04	1	0.04	0.04	0.13
7	0.04	0.24	0.07	0.04	0.07	0.03	0.04	1	0.13	0.13
8	0.14	0.29	0.12	0.14	0.08	0.18	0.04	0.13	1	0.07
9	0.08	0.08	0.01	0.09	0.21	0	0.13	0.13	0.07	1

TABLE III
CONFUSION MATRIX BY USING THE POOLING FEATURES.

C	0	1	2	3	4	5	6	7	8	9
0	979	0	0	0	0	0	0	1	0	0
1	0	1133	1	0	0	0	1	0	0	0
2	1	1	1025	0	0	0	0	5	0	0
3	0	0	0	1008	0	2	0	0	0	0
4	0	0	0	0	974	0	0	0	1	7
5	2	0	0	3	0	884	1	0	1	1
6	5	2	0	0	1	3	946	0	1	0
7	0	1	1	4	0	0	0	1019	1	2
8	2	0	2	1	0	1	0	1	964	3
9	1	1	0	2	3	2	0	1	3	996

TABLE IV
CONFUSION MATRIX BY USING HORIZONTAL DISPLACEMENT FEATURES (DIS-X).

C	0	1	2	3	4	5	6	7	8	9
0	976	0	0	0	0	0	1	1	2	0
1	0	1132	1	0	0	1	1	0	1	0
2	2	3	1016	1	1	0	0	6	3	0
3	0	0	0	1004	0	4	0	1	1	0
4	1	0	0	0	976	0	0	0	2	3
5	1	0	0	2	0	887	1	0	1	0
6	9	2	0	1	1	5	938	0	2	0
7	0	2	3	1	0	0	0	1018	2	2
8	2	0	1	2	0	1	0	1	966	1
9	2	3	0	1	5	8	0	4	6	980

TABLE V
CONFUSION MATRIX BY USING VERTICAL DISPLACEMENT FEATURES (DIS-Y).

C	0	1	2	3	4	5	6	7	8	9
0	975	0	1	0	0	0	1	2	1	0
1	0	1132	1	1	0	0	0	0	1	0
2	1	0	1026	0	0	0	0	4	1	0
3	0	0	0	1003	0	4	0	1	2	0
4	0	0	1	0	973	0	1	0	0	7
5	2	0	0	4	0	885	1	0	0	0
6	5	3	0	1	1	3	944	0	1	0
7	0	2	4	2	0	0	0	1018	0	2
8	2	0	1	1	0	1	0	3	964	2
9	1	1	0	3	4	1	0	1	3	995

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich *et al.*, "Going Deeper with Convolutions," in *CVPR*, 2015.

[4] S. R. Bulo, G. Neuhold, and P. Kontschieder, "Loss Maxpooling for Semantic Image Segmentation," in *CVPR*, 2017.

[5] X. Yu, J. Yang, T. Wang, and T. Huang, "Key Point Detection by Max Pooling for Tracking," *IEEE transactions on cybernetics*, 2015.

[6] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial Transformer Networks," in *NIPS*, 2015.

[7] L. v. d. Maaten and G. Hinton, "Visualizing Data using t-SNE," *JMLR*, 2008.

[8] D. Ciresan and U. Meier, "Multi-column Deep Neural Networks for Offline Handwritten Chinese Character Classification," in *IJCNN*, 2015.

[9] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: A Simple Deep Learning Baseline for Image Classification?" *TIP*, 2015.

[10] I.-J. Kim and X. Xie, "Handwritten Hangul Recognition Using Deep Convolutional Neural Networks," *IJDAR*, 2015.

[11] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Convolutional Neural Network Committees for Handwritten Character Classification," in *ICDAR*, 2011.

[12] C. Wu, W. Fan, Y. He, J. Sun, and S. Naoi, "Handwritten Character Recognition by Alternately Trained Relaxation Convolutional Neural Network," in *ICFHR*, 2014.

[13] F. Yin, Q.-F. Wang, X.-Y. Zhang, and C.-L. Liu, "ICDAR 2013 Chinese Handwriting Recognition Competition," in *ICDAR*, 2013.

[14] Y. Zheng, Y. Cai, G. Zhong, Y. Chherawala, Y. Shi, and J. Dong, "Stretching Deep Architectures for Text Recognition," in *ICDAR*, 2015.

[15] M. Zeiler and R. Fergus, "Stochastic Pooling for Regularization of Deep Convolutional Neural Networks," in *ICLR*, 2013.

[16] M. Malinowski and M. Fritz, "Learning Smooth Pooling Regions for Visual Recognition," in *BMVC*, 2013.

[17] N. Murray and F. Perronnin, "Generalized Max Pooling," in *CVPR*, 2014.

[18] R. Qian, Y. Yue, F. Coenen, and B. Zhang, "Traffic Sign Recognition with Convolutional Neural Network Based on Max Pooling Positions," in *ICNC-FSKD*, 2016.

[19] I. Goodfellow, H. Lee, Q. V. Le, A. Saxe, and A. Y. Ng, "Measuring Invariances in Deep Networks," in *NIPS*, 2009.

[20] D. Scherer, A. Müller, and S. Behnke, "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition," in *ICANN*, 2010.

[21] Y.-L. Boureau, J. Ponce, and Y. LeCun, "A Theoretical Analysis of Feature Pooling in Visual Recognition," in *ICML*, 2010.

[22] X. Wang, L. Wang, and Y. Qiao, "A Comparative Study of Encoding, Pooling and Normalization Methods for Action Recognition," in *ACCV*, 2012.

[23] P. Sermanet, S. Chintala, and Y. LeCun, "Convolutional Neural Networks Applied to House Numbers Digit Classification," in *ICPR*, 2012.

[24] Y. Igarashi and K. Fukui, "3D Object Recognition Based on Canonical Angles between Shape Subspaces," in *ACCV*, 2010.