Computational Visual Media
https://doi.org/10.1007/s41095-020-0181-9 Vol. 6, No. 4, December 2020, 467–476
Research Article

Kernel-blending connection approximated by a neural network for image classification

Xinxin Liu[1], Yunfeng Zhang[1] (✉), Fangxun Bao[2], Kai Shao[1], Ziyi Sun[1], and Caiming Zhang[1,2]

Abstract This paper proposes a kernel-blending connection approximated by a neural network (KBNN) for image classification. A kernel mapping connection structure, guaranteed by the function approximation theorem, is devised to blend feature extraction and feature classification through neural network learning. First, a feature extractor learns features from the raw images. Next, an automatically constructed kernel mapping connection maps the feature vectors into a feature space. Finally, a linear classifier is used as an output layer of the neural network to provide classification results. Furthermore, a novel loss function involving a cross-entropy loss and a hinge loss is proposed to improve the generalizability of the neural network. Experimental results on three well-known image datasets illustrate that the proposed method has good classification accuracy and generalizability.

Keywords image classification; blending neural network; function approximation; kernel mapping connection; generalizability

1 Introduction

Image classification assigns images to predefined categories by recognizing a subject or an object in images. Image classification is a classic image processing tool that is a basis for tasks such as image segmentation, behavior analysis, scene understanding, and other high-level visual tasks, and it has a wide range of practical applications, such as target recognition, object tracking, and image retrieval. Based on the feature extraction approach used, existing image classification methods can be broadly classified as prior-based methods and learning-based methods.

Prior-based image classification methods first extract image features according to empirical knowledge and then use a classifier. The support vector machine (SVM) [1] is a widely used classifier due to its good generalizability. In particular, a kernel-based SVM can deal effectively with nonlinear and high-dimensional data, and performs well on many image classification tasks. An incremental SVM that used histogram of oriented gradients (HOG) features as training vectors was proposed in Ref. [2] to classify images under different imaging conditions. In Ref. [3], using spectral features, an SVM-based sequential classifier was developed to classify multitemporal remote sensing images. Although such prior-based methods coupled with SVM classifiers show good classification performance, their feature extractors are hand crafted, so require domain knowledge. Thus, they are unsuitable for new data and tasks. Moreover, they cannot fully express the information in the raw images [4, 5].

Learning-based methods learn features automatically and directly from raw pixels. LeCun et al. [6] successfully applied a convolutional neural network (CNN) to handwritten character recognition and achieved remarkable classification performance; since then, CNNs have been widely applied to image classification tasks. In Ref. [7], a multimodal CNN was used to classify RGB-D images. Through convolution and pooling, color and depth features were fused effectively to maintain good classification

1 Shandong University of Finance and Economics, Jinan 250014, China. E-mail: X. Liu, liuxxin26@163.com; Y. Zhang, yfzhang@sdufe.edu.cn (✉); K. Shao, shaokai17862921498@126.com; Z. Sun, 17862921505@163.com.
2 Shandong University, Jinan 250100, China. E-mail: F. Bao, fxbao@sdu.edu.cn; C. Zhang, czhang@sdu.edu.cn.
Manuscript received: 2020-03-31; accepted: 2020-05-18

performance on images with significant noise or object occlusions. Based on a deep CNN, a fine-grained image classifier with generalized large-margin loss was proposed in Ref. [8], with improved classification performance on fine-grained images. Such CNN-based image classification methods extract salient features from raw images that are invariant to shifting or to shape distortions, but the algorithm used to train the CNN is based on empirical risk minimization—it attempts to minimize errors for the training set. However, with structural risk minimization, such models are less generalizable than SVMs [9], which aim to minimize generalization errors on unseen data given a fixed distribution for the training set.

In recent years, combining a CNN with an SVM for image classification has attracted considerable attention. In Ref. [10], a trained CNN was first applied to extract features from functional magnetic resonance images, and then an SVM was employed for classification. Experiments showed that this combination of SVM and CNN performed better than other classifiers. In Ref. [9], a hybrid model integrating a CNN and an SVM was used to recognize handwritten digits. The CNN functioned as a trainable feature extractor and the SVM was used as a classifier. These papers show that a combination of a CNN and an SVM for image classification can perform well. However, the CNN and SVM were trained separately. Thus, the SVM classification results could not provide feedback to assist in CNN training. Feature extraction and feature classification did not interact effectively, affecting classification accuracy.

These combined methods compensate for the limits of both CNNs and SVMs by incorporating the merits of both. However, they are based on two different algorithm architectures, which are inappropriate for a "hard connection". Seeking a mapping between CNN and SVM to establish a "soft connection" enables more flexible interaction. Furthermore, it is crucial to establish a precise mapping. To better blend feature extraction with feature classification for improved classification performance, one should establish an effective and precise mapping connection on a theoretical basis.

In this paper, a kernel-blending connection approximated by a neural network (KBNN) is proposed for neural networks applied to image classification tasks. Considering that a three-layered neural network with nonlinear units in the hidden layer can approximate both continuous and other kinds of functions, we devise a network module that can learn the kernel function for the SVM. Using this kernel mapping connection, which carries a theoretical guarantee, feature extraction and feature classification are blended organically and precisely. First, the image features are automatically learned by a feature extractor. Second, the extracted feature vector is mapped into a feature space through a kernel mapping connection module. Finally, the classification results are obtained using a linear classification layer. To further improve the KBNN's generalizability, we propose a novel loss function to train the network in which a hinge loss is introduced to the cross-entropy loss. The main contributions of this paper are as follows:

- A novel image classification method based on a new deep neural network, in which an SVM kernel function is learned through a subnetwork to blend a CNN and an SVM in a unified framework, providing improved classification performance.
- Inspired by the function approximation ability of neural networks, a kernel mapping connection that organically blends feature extraction with feature classification. Unlike traditional combination methods, this kernel mapping connection provides a soft connection between the CNN and SVM as it is performed as a component of the neural network. Furthermore, unlike traditional manual selection, the kernel mapping can be trained adaptively without use of kernel tricks to improve classification accuracy. This kernel mapping has a sound theoretical basis.
- A novel loss function for improved generalizability of the method. Unlike in traditional cross-entropy loss, a hinge loss is combined to minimize both empirical and structural risk.

2 Preliminaries

Inspired by the biological architecture of the mammalian visual cortex [11, 12], CNNs are characterized by limited receptive fields, shared weight parameters, and pooling layers [6]. This architecture allows CNNs to suppress increases in the number of weight parameters and makes them robust to parallel shifts of objects in images. The back-propagation algorithm [6] is generally employed in

CNNs to update the weight parameters by calculating the gradient obtained at the output layer and then backpropagating it to the input layer.

In recent years, CNNs have been successfully applied to practical situations and have made significant achievements in image processing [13, 14]. Zhang et al. [15] proposed a learning-based method to automatically detect and localize visual distractions in video. Video frames with extracted feature maps are first used as input layers for the network. Then, a state-of-the-art image segmentation CNN network, the end-to-end deep NN model SegNet, predicts a distraction map for every video frame; these are further refined in a post-processing step. Experimental results show that this method can efficiently improve the visual quality of video. Targeting the problem that conventional graph convolution methods fail to capture higher order information, Wen et al. [16] presented a motif-based graph convolution with variable temporal dense blocks for skeleton-based action recognition. It effectively fuses information from the different semantic roles of physically connected and disconnected joints to learn higher-order features. Furthermore, to enhance its ability to extract global temporal features, a non-local block is applied to capture whole-range dependencies in an attention mechanism. Experimental results on two challenging large-scale datasets demonstrate the effectiveness of the method.

3 Our method

This paper focuses on creating a new neural network to improve the image classification performance by blending feature extraction and feature classification in a unified framework that can be jointly trained. To this end, motivated by the function approximation ability of neural networks, we introduce a theoretically guaranteed kernel mapping connection module that provides a soft connection between feature extraction and feature classification. This is achieved by using a neural network to learn the kernel functions for the classifier.

The framework of the proposed KBNN network consists of three parts: feature extraction, kernel mapping connection, and feature classification, as shown in Fig. 1. First, a feature extraction subnetwork is applied to extract the input image features into a one-dimensional feature vector. This subnetwork uses a series of convolutional and pooling operations combined in several ways. Then, to enable a soft connection from feature extraction to feature classification, a kernel mapping connection module puts the extracted feature vector into a feature space; the result serves as the kernel function in the classifier. Finally, a linear classification layer is employed to classify the features in the feature space, giving the final results. To improve the generalizability of the network, a novel loss function with hinge loss is applied to train the neural network until convergence.

3.1 Feature extraction

To improve feature extraction performance, we use a feature extraction subnetwork with a series of convolutional and pooling operations combined using several techniques. The feature extraction subnetwork is composed of three convolutional layers (some of which are followed by pooling layers) and a fully connected layer. The convolutional layers are mainly used to extract feature maps using convolutional filters followed by a nonlinear activation

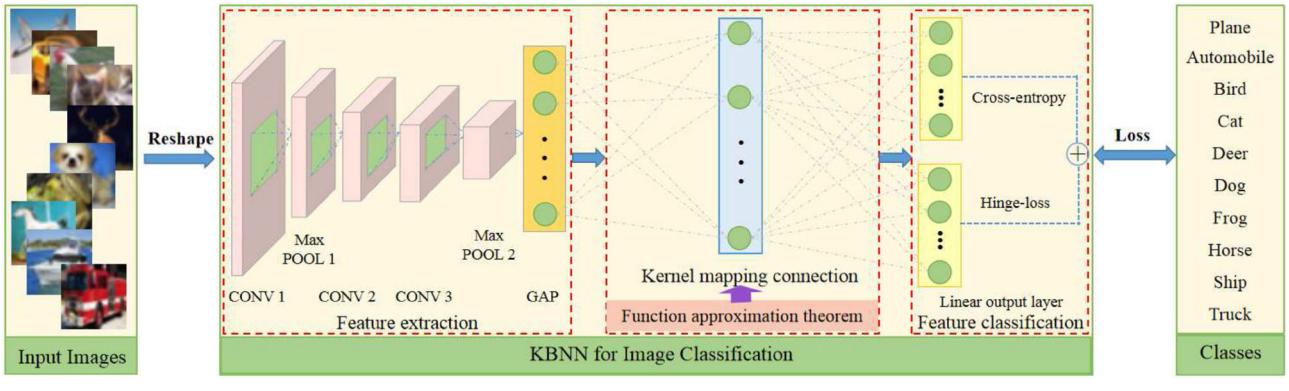

Fig. 1 Flowchart of our KBNN image classification method.

function; we adopt the rectified linear units (ReLU) function to avoid the vanishing gradient problem. To accelerate the training process, we employ batch normalization [17] before ReLU activation. The pooling layers group the local features from adjacent pixels. Max pooling and average pooling are adopted in different pooling layers. The fully connected layer integrates local feature information into a one-dimensional feature vector. To prevent overfitting in the traditional fully connected layer, we adopt global average pooling (GAP) [18], which takes the average of each feature map. Overfitting is avoided as this approach does not require parameter optimization. Moreover, because spatial information is summed out, this approach is more robust to spatial translations of features in the input images.

3.2 Kernel mapping connection

In most image classification methods based on combining a CNN and an SVM, feature vectors are extracted from the trained CNN and then input into the SVM classifier separately, because the CNN and SVM have different implementation frameworks. Therefore, the connection between the CNN and the SVM is a "hard connection": feature extraction and feature classification are trained separately and do not interdepend. To integrate feature extraction and feature classification into an organic whole, it is necessary to use a "soft connection". We regard the SVM kernel function as the point at which to address this problem, by applying the function approximation ability of a neural network. The kernel function in an SVM is continuous and can realize linear separability by mapping the linearly inseparable space into a higher dimensional feature space. Cybenko [19] has proved theoretically that a three-layer neural network can approximate any continuous nonlinear function arbitrarily well on a compact interval. Inspired by this function approximation theorem, we devise a kernel mapping connection that learns the kernel function using a neural network, to enable a soft connection between feature extraction and feature classification. The theorem is given as follows:

Theorem 1. Let I_n be the n-dimensional unit cube, $[0,1]^n$. Let $C(I_n)$ denote the space of continuous functions on I_n, and $M(I_n)$ denote the space of finite and signed regular Borel measure on I_n. Let σ be any continuous sigmoidal function. Let $Y_j, X \in \Re^n$, and θ_j,$\alpha_j \in \Re$. Then, finite sums of the form:

$$G(X) = \Sigma_{j=1}^{N} \alpha_j \sigma(Y^{\mathrm{T}} X + \theta_j)$$

are dense in $C(I_n)$. In other words, given any $f \in C(I_n)$ and $\varepsilon > 0$, there is a sum, $G(x)$, of the above form, for which:

$$|G(X) - f(X)| < \varepsilon, \quad \forall X \in I_n$$

Based on this function approximation theorem, a kernel mapping connection is established to map the feature vectors from the GAP layer to a feature space used as the input to the linear classification layer.

As shown in Fig. 2, $D = (x_1, \cdots, x_m)$, $x_i \in \Re$ is the feature vector output by the GAP layer with d neurons. The kernel mapping layer contains q neurons, and the kernel mapping output layer has l neurons. The kernel mapping learned by a neuron, that is, the input of a neuron y_k in the linear output layer is

$$y_k = \sum_{m=1}^{q} \eta_{km} \sigma \left(\sum_{i=1}^{d} \nu_{mi} x_i + \beta_m \right) \tag{1}$$

where $1 \leqslant i \leqslant d$, $1 \leqslant m \leqslant q$, $1 \leqslant k \leqslant l$, ν_{mi}, η_{km} are the weight vectors, β_m is the bias of the mth neuron, and $\sigma(\cdot)$ is the sigmoid function.

3.3 Feature classification

Unlike the softmax layer minimizing cross-entropy loss used in traditional neural networks, to improve the generalizability of the proposed network, our novel loss function combines cross-entropy loss and hinge loss to minimize both empirical and structural risk.

The traditional softmax loss function is

$$J_{\mathrm{s}} = -\frac{1}{N} \sum_{i=1}^{M} \sum_{j=1}^{K} p_{i,j}^{-} \log(p_{i,j}) \tag{2}$$

where $i = 1, \ldots, M$, $j = 1, \ldots, K$, M and K are the numbers of training images and classes, respectively, and $p_{i,j}$ denotes the probability between the image X_i in class j and the ground truth. After introducing the

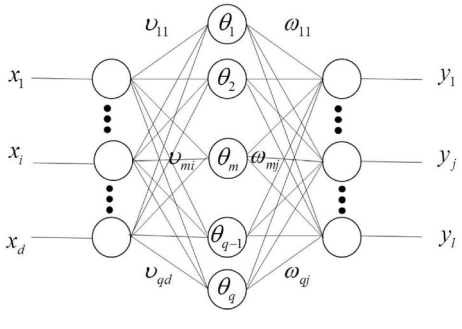

Fig. 2 Kernel mapping connection.

positive penalty factor C for the SVM, the improved hinge loss is

$$J_{\mathrm{h}} = C\sum_{i=1}^{n}\max\left(0, 1 - y_i(\omega^{\mathrm{T}}x_i + b)\right)^2 \quad (3)$$

where C controls the tradeoff between maximizing the margin and misclassification, ω is the weight vector, and b is the bias. By combining the cross-entropy loss and the improved hinge loss, the proposed loss function is defined as follows:

$$J = J_{\mathrm{s}} + J_{\mathrm{h}} \quad (4)$$

When applying the loss function to train the KBNN, the weights and biases in the feature extraction layers and kernel mapping layer are learned by backpropagating the gradients from the linear classification layer.

4 Results and discussion

In this section, we report the results of a variety of experiments performed to evaluate the performance of the proposed KBNN image classification method.

4.1 Experimental details

We conducted experiments on three datasets: MNIST, CIFAR-10, and CIFAR-100 [20, 21]. These datasets are widely used and specifically intended for investigating the performance of image classification methods. MNIST is a handwritten digit dataset in which the goal is to classify handwritten numerals 0 to 9. The dataset contains 60,000 training images and 10,000 test images, each of which is a 28×28 pixel grayscale image. The CIFAR-10 dataset consists of 50,000 training images and 10,000 test images. Each image is a 32×32 RGB image that belongs to one of ten natural-object categories. In CIFAR-10, the object positions and scales within categories and their colors and textures between categories vary significantly. The CIFAR-100 dataset has the same size and format of images as the CIFAR-10 database, but contains 100 classes. Thus, the CIFAR-100 dataset only has one tenth as many labeled images in each class, i.e., 500 training images and 100 testing images.

To reveal the generality of our proposed method, we applied the same neural network architecture to all datasets, set up as shown in Table 1. We used the mini-batch gradient descent method to learn the parameters and adopted Adam to accelerate network convergence. For MNIST, the batch size and number of epochs were 128 and 400, respectively, and for CIFAR-10 they were 128 and 250, respectively. For CIFAR-100, we use the same settings as for CIFAR-10. The learning rate was initialized to 0.0001, and the weight decay was set to 0.0001. The KBNN model was implemented using TensorFlow. All experiments were conducted on a PC equipped with an NVIDIA GTX Titan X GPU.

Table 1 Configuration of KBNN architecture

Layer	Type	Kernel	Stride	Padding	Channels
Data	Input	N/A	N/A	N/A	N/A
CONV 1	Convolution	5×5	2	SAME	64
POOL 1	Average pooling	3×3	2	VALID	64
CONV 2	Convolution	3×3	2	SAME	128
CONV 3	Convolution	3×3	2	SAME	256
POOL 2	Max pooling	2×2	2	VALID	256
GAP	Average pooling	1×1	1	VALID	256
Kernel mapping	Fully connected	1×1	N/A	N/A	128
Linear output	Output	1×1	N/A	N/A	10

4.2 Loss function analysis

The penalty parameter C adopted in the proposed loss function controls the tradeoff between margin maximization and classification violation. We first analyzed the impact of this parameter on the performance of KBNN, by investigating variation of classification error for values of C in the range of 0.001–1000, using MNIST and CIFAR-10 datasets. The mean error over 5 independent trials is plotted in Fig. 3. When $C < 3$, low classification accuracy results for both datasets, particularly MNIST. As C increases from 4 to 10, classification accuracy improves, and for $C > 10$, classification accuracy

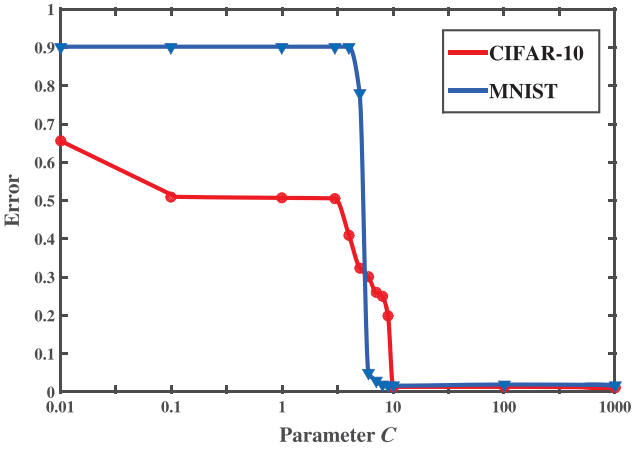

Fig. 3 Impact of penalty parameter C on error.

converges. As Fig. 3 shows, any arbitrary value for C of at least 10 is acceptable for the loss function. Thus, we set $C = 10$ in the remaining experiments.

Next, we compared the performance of the new loss function to those of cross-entropy loss and hinge loss. The results for MNIST and CIFAR-10 are presented in Figs. 4 and 5, respectively. On MNIST, with grayscale images, KBNN with the proposed loss function performs best regarding errors, while hinge loss is worst. Moreover, compared with cross-entropy loss and hinge loss, the KBNN loss function converges faster. On the more complex CIFAR-10 dataset, which is composed of RGB images, the proposed loss function is more stable, especially in comparison to cross-entropy loss. More generally, these results show that using a linear output layer with the improved loss function performs better than a traditional output layer with cross-entropy loss and hinge loss.

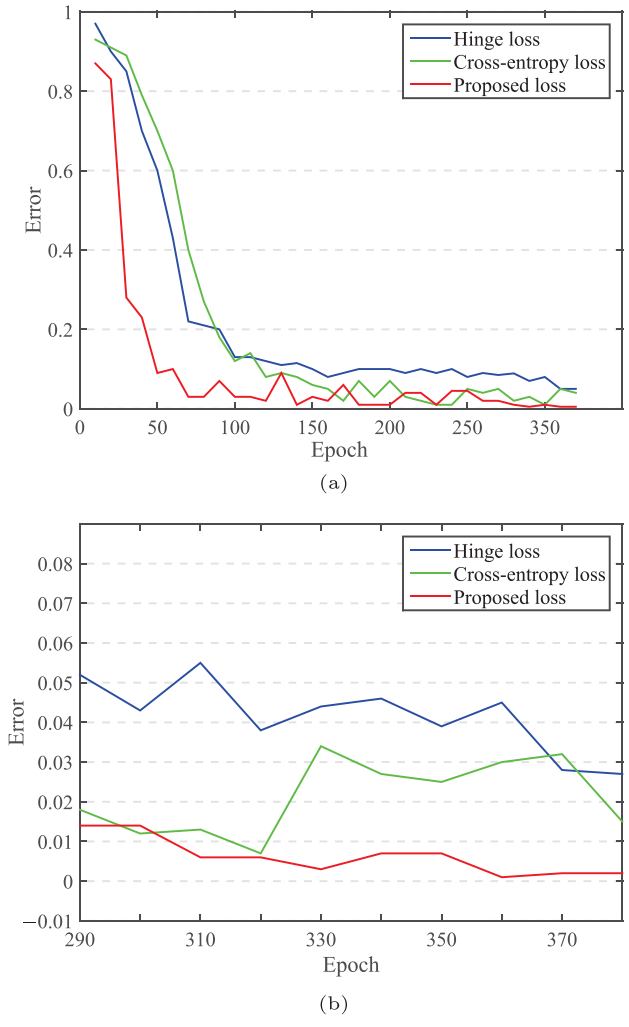

Fig. 4 Smoothed test error on MNIST for cross-entropy loss, hinge loss, and proposed loss: (a) epochs 0–400, (b) epochs 300–400.

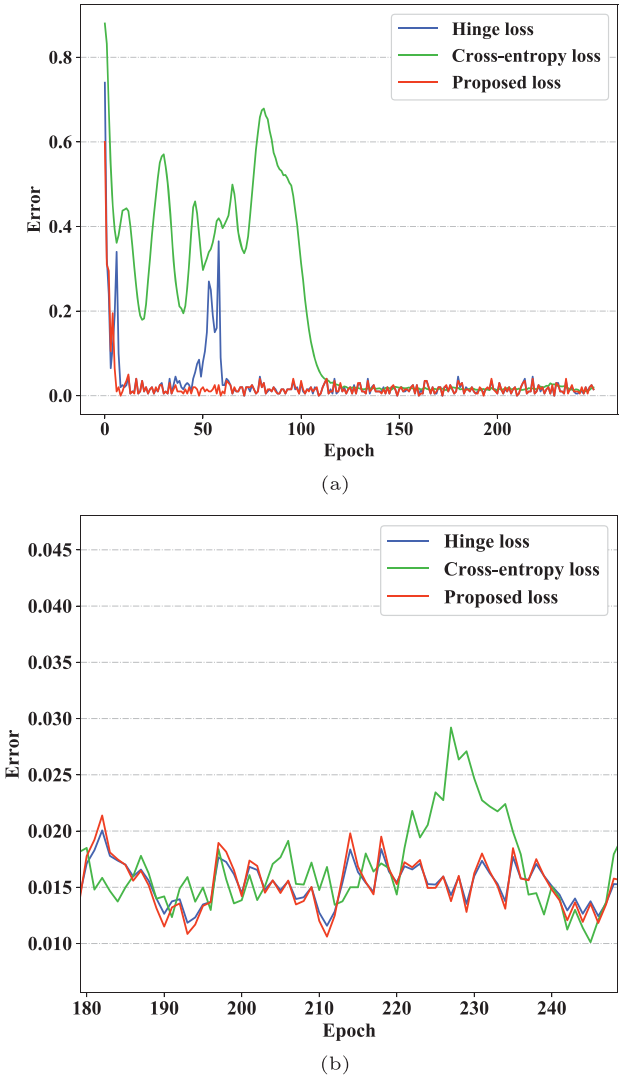

Fig. 5 Smoothed test error on CIFAR-10 for cross-entropy loss, hinge loss, and proposed loss: (a) epochs 0–250, (b) epochs 180–250.

4.3 Comparison with the state-of-the-art

We next compare the classification performances of the proposed KBNN and state-of-the-art methods for these three datasets.

4.3.1 MNIST results

To verify the effectiveness of the proposed KBNN image classification method, we first compared the proposed KBNN with state-of-the-art methods on the MNIST dataset, including two combined methods: DLSVM [22] and Niu and Suen's [9], a method using CNN with softmax (CNN+softmax) [23], and four other representative methods: CDBM [24], PCAnet [25], Deep NCAE [26], and Drplu [27]. All methods

were trained on the original training dataset except for Niu and Suen's method, which was trained with an augmented training dataset by using distortion techniques. Classification error results are summarized in Table 2. As can be seen, KBNN performs better on MNIST than most of the other methods in the comparison. In particular, KBNN achieves higher classification accuracy than the traditional combined method DLSVM, which indicates that the kernel mapping contributes to classification accuracy. The KBNN also outperforms CNN+softmax, further showing the good performance of the proposed loss function. Unlike Niu and Suen's method, which uses distortion techniques to augment the training dataset and enhance generalizability, KBNN is trained directly on the original training set, so KBNN's results are weaker than those of Niu and Suen's method. However, the difference in classification accuracy is quite small: the KBNN result trails Niu and Suen's method's by 0.17.

4.3.2 CIFAR-10 results

To further illustrate the generalizability of KBNN, we compared it to six other image classification methods using the CIFAR-10 dataset. These were: the combination method DLSVM, two improved loss function methods, large-margin Gaussian mixture loss (ResNst110+L-GM [23]) and multi-loss (ML-DNN [28]), and three high-performance methods: NIN [18], maxout networks [29], and drop-connect [30]. Classification error results are presented in Table 3. It can be seen that KBNN has lowest errors for CIFAR-10. In particular, the network architecture of KBNN suffices for more complex datasets, verifying that KBNN has good generalizability.

4.3.3 CIFAR-100 results

To investigate the performance of KBNN on a more complex dataset, we further compared KBNN with six representative low-error methods on the CIFAR-100 dataset: learned pooling [31], stochastic pooling [32], maxout networks, NIN, ML-DNN, and ResNet (110-layer) [33].

Table 4 gives classification errors for our proposed KBNN and these other methods. It can be seen that KBNN surpasses several methods with a test error of 32.71%. ResNet is an outstanding network for diverse applications, which uses deep residual learning to overcome the difficulty of training a deeper network. Although the classification errors of KBNN are larger than for RestNet, KBNN has fewer network layers: KBNN can obtain good performance with a relatively small model size. Generally, the experimental results of Table 4 show that KBNN is also competitive for more complex datasets.

5 Conclusions and future work

This study has proposed a novel deep neural network for image classification with an approximate theorem-based kernel blending connection. To implement a soft connection between a CNN and an SVM, we established a kernel mapping connection structure, guaranteed by the function approximation theorem, to better blend feature extraction and feature classification. Neural network learning further increases the adaptability of the connection, which avoids the need for kernel tricks as applied in traditional SVMs. Moreover, we combine a hinge loss with cross-entropy loss to improve the generalizability of KBNN.

In future research, we will focus on further improving the generalizability of KBNN in terms of network architecture optimization, including the number of layers and hidden neurons, the size of the convolution kernel, and the value of the penalty factor.

Table 2 Classification errors (%) on MNIST

Method	DLSVM	Niu and Suen's	CNN+softmax	CDBM	PCANet	Deep NCAE	Drplu	KBNN
Error	0.87	0.19	0.68	0.82	0.62	2.09	1.04	0.36

Table 3 Classification errors (%) using CIFAR-10

Method	DLSVM	ResNst110+L-GM	ML-DNN	NIN	Maxout Networks	Drop-Connect	KBNN
Error	11.9	4.96	8.12	8.81	9.38	9.32	1.54

Table 4 Classification errors (%) on CIFAR-100

Method	Stochastic Pooling	Learned Pooling	Maxout Networks	NIN	ML-DNN	ResNet	KBNN
Error	42.51	43.71	38.57	35.68	34.18	28.62	32.71

The experimental results on benchmark datasets indicate that, although KBNN shows promising classification performance and generalizability, the network architecture and the penalty factor still need to be set manually. Even though we performed a parameter sensitivity test on the penalty factor, upper and lower bounds on its value referred to empirical settings in other literature. Such empirical settings of parameters and architecture may affect the performance of the method on other datasets. Therefore, our further research work will focus on improving model generalizability. We will attempt to set the penalty factor as a trainable parameter and optimize the network architecture with intelligent optimization algorithms.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 61972227 and 61672018), the Natural Science Foundation of Shandong Province (Grant No. ZR2019MF051), the Primary Research and Development Plan of Shandong Province (Grant No. 2018GGX101013), and the Fostering Project of Dominant Discipline and Talent Team of Shandong Province Higher Education Institutions.

References

[1] Cortes, C.; Vapnik, V. Support-vector networks. *Machine Learning* Vol. 20, 273–297, 1995.

[2] Bagarinao, E.; Kurita, T.; Higashikubo, M.; Inayoshi, H. Adapting SVM image classifiers to changes in imaging conditions using incremental SVM: An application to car detection. In: *Computer Vision–ACCV 2009. Lecture Notes in Computer Science, Vol. 5996*. Zha, H.; Taniguchi, R.; Maybank, S. Eds. Springer Berlin Heidelberg, 363–372, 2010.

[3] Guo, Y. Q.; Jia, X. P.; Paull, D. Effective sequential classifier training for SVM-based multitemporal remote sensing image classification. *arXiv preprint* arXiv:1706.04719, 2017.

[4] Hinton, G. E.; Osindero, S.; Teh, Y. W. A fast learning algorithm for deep belief nets. *Neural Computation* Vol. 18, No. 7, 1527–1554, 2006.

[5] Bengio, Y.; Courville, A.; Vincent, P. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 35, No. 8, 1798–1828, 2013.

[6] LeCun, Y.; Boser, B. E.; Denker, J. S.; Henderson, D.; Howard, R.; Hubbard, W.; Jackel, L. D. Back propagation applied to handwritten zip code recognition. *Neural Computation* Vol. 1, No. 4, 541–551, 1989.

[7] Eitel, A.; Springenberg, J. T.; Spinello, L.; Riedmiller, M.; Burgard, W. Multimodal deep learning for robust RGB-D object recognition. *arXiv preprint* arXiv:1507.06821, 2015.

[8] Shi, W. W.; Gong, Y. H.; Tao, X. Y.; Cheng, D.; Zheng, N. N. Fine-grained image classification using modified DCNNs trained by cascaded softmax and generalized large-margin losses *IEEE Transactions on Neural Networks and Learning Systems* Vol. 30, No. 3, 683–694, 2018.

[9] Niu, X. X.; Suen, C. Y. A novel hybrid CNN–SVM classifier for recognizing handwritten digits *Pattern Recognition* Vol. 45, No. 4, 1318–1325, 2012.

[10] Sun, X.; Park, J.; Kang, K.; Hur J. Novel hybrid CNN-SVM model for recognition of functional magnetic resonance images. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 1001–1006, 2017.

[11] Hubel, D. H.; Wiesel, T. N. Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology* Vol. 195, No. 1, 215–243, 1968.

[12] Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics* Vol. 36, No. 4, 193–202, 1980.

[13] Zeiler, M. D.; Fergus, R. Visualizing and understanding convolutional networks. In: *Computer Vision – ECCV 2014. Lecture Notes in Computer Science, Vol. 8689*. Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T. Eds. Springer Cham, 818–833, 2014.

[14] Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. OverFeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint* arXiv:1312.6229, 2013.

[15] Zhang, F. L.; Wu, X.; Li, R.-L.; Wang, J.; Zheng, Z. H.; Hu, S. M. Detecting and removing visual distractors for video aesthetic enhancement. *IEEE Transactions on Multimedia* Vol. 20, No. 8, 1987–1999, 2018.

[16] Wen, Y. H.; Gao, L.; Fu, H. B.; Zhang, F. L.; Xia, S. H. Graph CNNs with motif and variable temporal block for skeleton-based action recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence Vol. 33, 8989–8996, 2019.

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

[18] Lin, M.; Chen, Q.; Yan, S. C. Network in network. *arXiv preprint* arXiv:1312.4400, 2013.

[19] Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems* Vol. 2, No. 4, 303–314, 1989.

[20] LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* Vol. 86, No. 11, 2278–2324, 1998.

[21] Krizhevsky, A.; Hinton, G. Learning multiple layers of features from tiny images. Master Thesis. University of Toronto, 2009.

[22] Tang, Y. Deep learning using support vector machines. *arXiv preprint* arXiv:1306.0239, 2015.

[23] Wan, W. T.; Zhong, Y. Y.; Li, T. P.; Chen, J. S. Rethinking feature distribution for loss functions in image classification. *arXiv preprint* arXiv:1803.02988, 2018.

[24] Lee, H.; Grosse, R.; Ranganath, R.; Ng, A. Y. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the 26th Annual International Conference on Machine Learning, 609–616, 2009.

[25] Chan, T. H.; Jia, K.; Gao, S. H.; Lu, J. W.; Zeng, Z. N.; Ma, Y. PCANet: A simple deep learning baseline for image classification? *IEEE Transactions on Image Processing* Vol. 24, No. 12, 5017–5032, 2015.

[26] Hosseini-Asl, E.; Zurada, J. M.; Nasraoui, O. Deep learning of part-based representation of data using sparse autoencoders with nonnegativity constraints. *IEEE Transactions on Neural Networks and Learning Systems* Vol. 27, No. 12, 2486–2498, 2016.

[27] Bristow, H.; Eriksson, A.; Lucey, S. Fast convolutional sparse coding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 391–398, 2013.

[28] Xu, C. Y.; Lu, C. Y.; Liang, X. D.; Gao, J. B.; Zheng, W.; Wang, T. J.; Yan, S. C. Multi-loss regularized deep neural network. *IEEE Transactions on Circuits and Systems for Video Technology* Vol. 26, No. 12, 2273–2283, 2016.

[29] Goodfellow, I. J.; Warde-Farley, D.; Mirza, M. Courville, A.; Bengio, Y. Maxout networks. *arXiv preprint* arXiv:1302.4389, 2013.

[30] Wan, L.; Zeiler, M.; Zhang, S.; LeCun, Y.; Fergus, R. Regularization of neural networks using dropconnect. In: Proceedings of the 30th International Conference on Machine Learning, Vol. 28, 1058–1066, 2013.

[31] Malinowski, M.; Fritz, M. Learnable pooling regions for image classification. *arXiv preprint* arXiv:1301.3516, 2013.

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

[33] He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778, 2016.

Xinxin Liu received her B.E. degree from the School of Computer Science and Technology, North China Institute of Science and Technology, Langfang, China, in 2016. She is currently working toward her M.S. degree at Shandong Provincial Key Laboratory of Digital Media Technology, Shandong University of Finance and Economics. Her research interests include particle swarm optimization, machine learning, and image processing.

Yunfeng Zhang received his B.E. degree in computational mathematics and application software from Shandong University of Technology, Jinan, China, in 2000, his M.S. degree in applied mathematics from Shandong University in 2003, and his Ph.D. degree in computational geometry from Shandong University in 2007. He is now a professor in Shandong Provincial Key Laboratory of Digital Media Technology, Shandong University of Finance and Economics. His current research interests include computer-aided geometric design, digital image processing, computational geometry, and function approximation.

Fangxun Bao received his M.Sc. degree from the Department of Mathematics of Qufu Normal University, China, in 1994, and his Ph.D. degree from the Department of Mathematics of Northwest University, Xi'an, China, in 1997. His current position is full professor in the Department of Mathematics, Shandong University. His research interests include computer-aided geometric design and computation, computational geometry, and function approximation.

Kai Shao received his B.E. degree from the School of Computer Science and Technology at Shandong University of Finance and Economics in 2018. He is currently working toward his M.S. degree at Shandong Provincial Key Laboratory of Digital Media Technology, Shandong University of Finance and Economics. His research interests include medical image processing and deep learning.

Ziyi Sun received her B.E. degree from the School of Computer Science and Technology at Shandong University of Finance and Economics in 2018. She is currently working toward her M.S. degree at Shandong Provincial Key Laboratory of Digital Media Technology, Shandong University of Finance and Economics. Her research interests include image processing and deep learning.

Caiming Zhang is a professor and doctoral supervisor of the School of Computer Science and Technology at Shandong University. He is now also the dean and professor of the School of Computer Science and Technology at Shandong Economic University. He received his B.S. and M.E. degrees in computer science from Shandong University in 1982 and 1984, respectively, and his Dr.Eng. degree in computer science from Tokyo Institute of Technology, Japan, in 1994. From 1997 to 2000, he held a visiting position at the University of Kentucky, USA. His research interests include CAGD, CG, information visualization, and medical image processing.