Computational Visual Media
DOI 10.1007/s41095-xxx-xxxx-x Vol. x, No. x, month year, xx–xx

Article Type (Research/Review)

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

Xinxin Liu[1], **Yunfeng Zhang**[1](✉), Fangxun Fangxun[2], Kai Shao[1], Ziyi Sun[1], and Caiming Zhang[1,3]

Abstract This paper proposes a kernel-blending connection approximated by a neural network (KBNN) for image classification, where 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 is designed to learn features from the raw images. Then, a kernel mapping connection is automatically constructed to map the feature vectors into a feature space. Finally, the linear classifier is conducted as an output layer of the neural network to obtain classification results. Furthermore, a novel loss function involving a cross-entropy loss and a hinge loss is proposed to improve the generalization ability of the neural network. Experimental results on three well-known image datasets illustrate that the proposed method has good classification accuracy and generalization ability.

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 technology 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. In general, based on the feature extraction approach, existing image classification methods can be broadly classified into two main categories: prior-based methods and learning-based methods.

Prior-based image classification methods first extract image features according to empirical knowledge and then choose a classifier to perform classification. 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 such models perform well on many image classification tasks. An incremental SVM that used HOG features as training vectors was proposed in [2] to classify images under different imaging conditions. In [3], based on spectral features, an SVM-based sequential classifier was developed to classify multitemporal remote sensing images. Although these prior-based methods coupled with SVM classifiers show good classification performance, their feature extractors are hand crafted, which requires domain knowledge; thus, they are not suitable for new data and tasks. Moreover, they cannot fully express the information in the raw images [4, 5].

Learning-based methods learn feature automatically directly from raw pixels. Yan 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 [7], a multimode CNN was used to classify RGB-D images. Through convolution and pooling,

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: fxbao@sdu.edu.cn.

3 Shandong University, Jinan, 250101, China. E-mail: czhang@sdu.edu.cn.

Manuscript received: 2014-12-31; accepted: 2015-01-30.

the color and depth features were fused effectively to maintain good classification performance on images containing more noise or object occlusions. Based on a deep CNN, a fine-grained image classifier with generalized large-margin loss was proposed in [8] to improve the classification performance on fine-grained images. These CNN-based image classification methods extract salient features from raw images that are invariant to shifting or to shape distortions, but the algorithm to train a CNN is based on empirical risk minimization—it attempts to minimize the errors in the training set. However, with the structural risk minimization, this model is less generalizable than SVM [9], which aims to minimize the generalization errors on the unseen data with a fixed distribution for the training set.

In recent years, the combination of a CNN and an SVM for image classification has attracted considerable attention. In [10], a trained CNN was first applied to extracting features from functional magnetic resonance images; then, an SVM was employed for classification. The experiments showed that compared with other classifiers, the combination of SVM and CNN achieved the best classification performance. In [9], a hybrid model integrating of CNN and SVM was presented for recognizing handwritten digits in which the CNN functioned as a trainable feature extractor and the SVM was used as a classifier. These methods verified that a combination of a CNN and an SVM can achieve good performance in image classification; however, the CNN and SVM were trained separately; thus, the SVM classification results could not act as timely feedback to assist in the CNN training. The result was that feature extraction and feature classification did not interact effectively, which affected the classification accuracy.

These combination methods compensate for the limits of both CNNs and SVMs by incorporating the merits of both classifiers. However, they are based on two different algorithm architectures, which is not appropriate for a "hard connection". Seeking a mapping between CNN and SVM to establish a "soft connection" enables more flexible interactions. Furthermore, it is crucial to establish a precise mapping. To better blend feature extraction with feature classification to improve the classification performance, it is necessary to 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 continuous or 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 through a linear classification layer. To further improve 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:

(1) We propose a novel image classification method based on a new deep neural network, in which the SVM kernel function is learned through a subnetwork to blend the CNN and the SVM in a unified framework to improve the classification performance.

(2) Inspired by the function approximation ability of the neural network, a kernel mapping connection is presented that organically blends feature extraction with feature classification. As opposed to traditional combination methods, the kernel mapping connection enables a soft connection between the CNN and SVM because it is performed as a component of the neural network. Furthermore, compared with traditional manual selection, the kernel mapping can be trained adaptively without the use of kernel tricks to improve the classification accuracy. Moreover, the establishment of kernel mapping is ensured on a theoretical basis.

(3) A novel loss function is developed to improve the generalizability of the method. In contrast to traditional cross-entropy loss, a hinge loss is introduced 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 distractors in videos. Video frames with extracted feature maps are first used as input layers for the network. Then, an end-to-end deep neural network model SegNet, which is a state-of-the-art image segmentation CNN network, is chosen to predict a distractor map in every video frame. Finally, the detection results are further refined in a post-processing step. Experimental results show that this method can efficiently improve the visual quality of videos. Aiming at the problem that conventional graph convolution methods fail to capture highorder information, Wen et al. [16] presented a motif-based graph convolution with variable temporal dense block for skeleton-based action recognition, which effectively fuse information from different semantic roles of physically connected and disconnected joints to learn high-order features. Furthermore, to enhance the ability for extracting global temporal features, a non-local block is applied to capture whole-range dependencies in an attention mechanism. The experimental results on two challenging large-scale dataset 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 trained together. To this end, motivated by the function approximation ability of neural networks, we introduce a theoretically guaranteed kernel mapping connection module that enables 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, to extract image features, a feature extraction subnetwork is applied to extract the input image features into a one-dimensional feature vector. This subnetwork design uses a series of convolutional and pooling operations combined with several techniques. Then, to enable a soft connection from feature extraction to feature classification, a kernel mapping connection module is established to convert 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 and obtain the final classification results. More specifically, to improve the generalizability of the network, a novel loss function introduced by hinge loss is applied to train the neural network until the training process converges.

3.1 Feature extraction

To improve the feature extraction performance, we design a feature extraction subnetwork with a series of convolutional and pooling operations combined with 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 function; here, we adopt the 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 at this layer because this approach does not require parameter optimization. Moreover, because the spatial information is summed out, this approach is more robust for spatial translations of features in the input images.

3.2 Kernel mapping connection

In most image classification methods based on a CNN and SVM combination, the 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 CNN and SVM is a “hard connection” in which feature extraction and feature classification are trained separately and do not effectively interact. To integrate feature extraction and feature classification into an organic whole, it is necessary to find 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 SVM is a type of continuous kernel function that

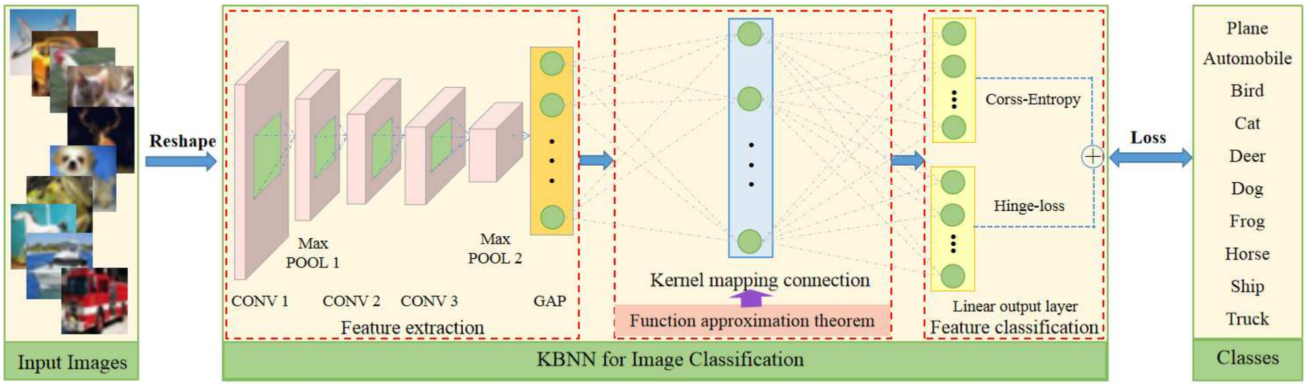


Fig. 1 Flowchart of the KBNN image classification method.

can realize linear separability by mapping the linear indivisible space into a higher dimensional feature space. Cybenko [19] 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, $[0,1]^n$. $C(I_n)$ denotes the space of continuous functions on I_n, and $M(I_n)$ denotes the space of finite and signed regular Borel measure on I_n. Let σ be any continuous sigmoidal function. $Y_j, X \in \Re^n, \theta_j, \alpha_j \in \Re$. Then, finite sums of the form

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

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

$$|G(X) - f(X)| < \varepsilon \quad for\ all\ X \ \in I_n. \qquad (2)$$

Based on the function approximation theorem in the theory of neural networks, 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, $D = (x_1, x_2, \ldots,\ x_m)\ x_i \in \Re$ 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

$$y_k = \sum_{m=1}^{q} \eta_{km} \sigma(\sum_{i=1}^{d} \nu_{mi} x_i + \beta_m), \qquad (3)$$

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

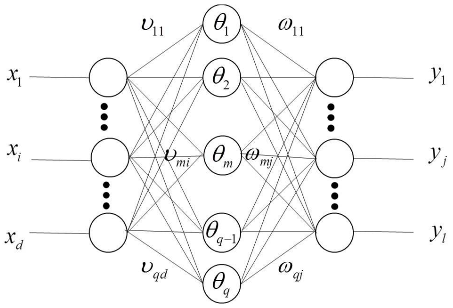


Fig. 2 Diagram of the kernel mapping connection.

3.3 Feature classification

Different from the softmax layer minimizing cross-entropy loss in traditional neural networks, to improve the generalizability of the proposed network, we propose a novel loss function that combines cross-entropy loss and hinge loss to minimize both empirical and structural risk.

The traditional softmax loss function is

$$J_{softmax} = -\frac{1}{N}\sum_{i=1}^{M}\sum_{j=1}^{K} p_{i,j}^{-} log(p_{i,j}), \qquad (4)$$

where $i = 1, \cdots, M, j = 1, \cdots, K$, 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 positive penalty factor C for the SVM, the improved hinge loss is

$$J_{hinge} = C\sum_{i=1}^{n} \max(0, 1 - y_i(\omega^T x_i + b))^2, \qquad (5)$$

where C controls the tradeoff between maximizing the margin and misclassification, ω is the weight vector,

Tab. 1 Configuration of KBNN architecture on MNIST and CIFAR-10.

Layer	Type	Kernel	Stride/Padding	Number of channel
data	Input	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	Full connected	1 × 1	N/A	128
Linear Output	Output	1 × 1	N/A	10

and b is the bias. By combining the cross-entropy loss and the improved hinge loss, the proposed loss function is defined as follows:

$$J = J_{softmax} + J_{hinge}. \tag{6}$$

When applying the loss function to train 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

We report the results of a variety of experiments in this section performed to evaluate the performance of the proposed KBNN image classification method.

4.1 Implementation details

We conducted experiments on three datasets: MNIST [20], CIFAR-10 [21] and CIFAR-100 [21]. Both these datasets are widely used and specifically intended for investigating the performance of image classification methods. MNIST is a handwritten digit classification dataset in which the goal is to classify handwritten numerals from $0, \ldots 9$. 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 hand format of images as the 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 both datasets, as presented in Tab. 1. We used the mini-batch gradient descent method to learn the parameters and adopted Adam to accelerate network convergence. In MNIST, the batch size and number of epochs are 128 and 400, respectively, and in CIFAR-10, they are 128 and 250, respectively. For CIFAR-100, we use the same setting as the CIFAR-10 dataset. 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 the experiments were conducted on a PC equipped with an NVIDIA GTX Titan X GPU.

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 analyze the impact of this parameter on the performance of KBNN. In the experiments, we investigated the classification error of KBNN based on the value of the penalty parameter C in the range of 0.001 to 1000 on the MNIST and CIFAR-10 datasets. The implementation details were the same as those described in Section 4.1. The results of the mean errors of 5 independent trials are reported in Fig. 3.

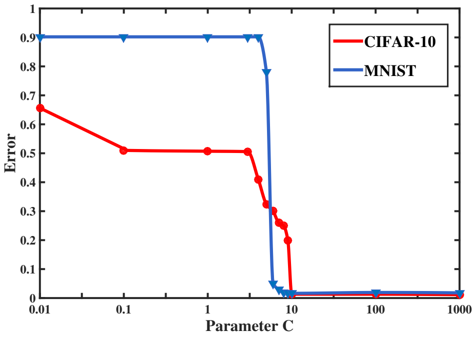
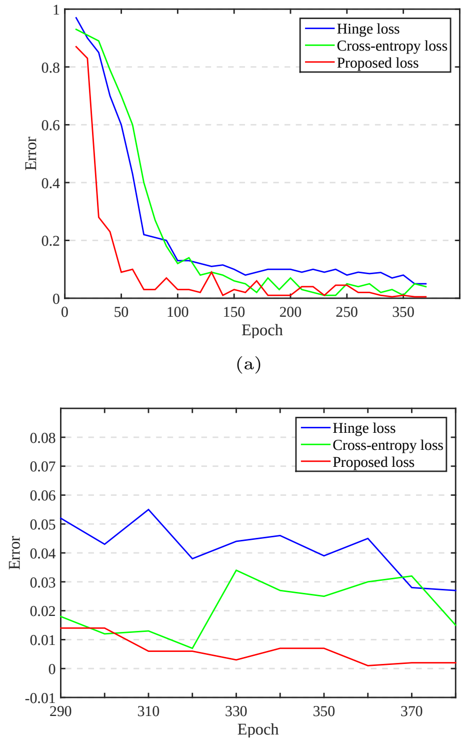


Fig. 3 Impact of parameter C on the error value.

In Fig. 3, when $C < 3$, low classification accuracies can be observed for both datasets, particularly for MNIST. As the value of C increases from 4 to 10, the classification accuracies improve, and when $C > 10$, the classification accuracies tend to converge. As Fig. 3 shows, an arbitrary value greater than 10 of C is acceptable for the loss function. Thus, we adopted 10 as the value of the penalty parameter C for the remaining experiments.

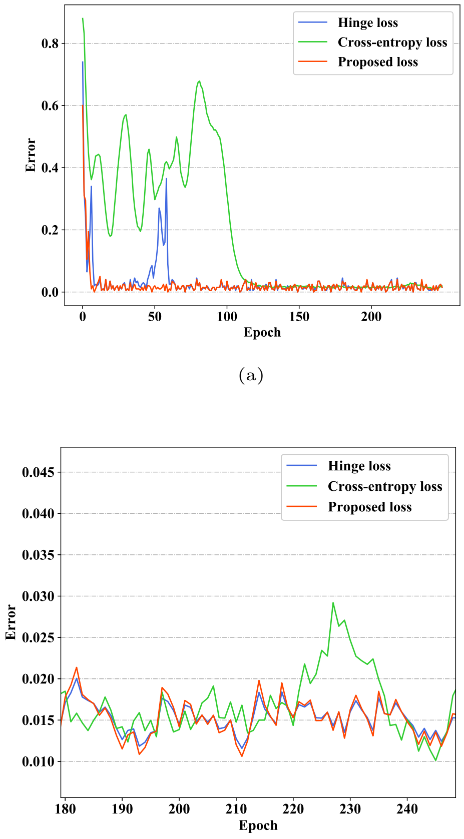


(a)

(b)

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

Next, we evaluate the performance of the improved loss function compared with cross-entropy loss and hinge loss. The results of the test error curves on MNIST and CIFAR-10 are presented in Fig. 4 and Fig. 5, respectively. On MNIST, with grayscale images, the KBNN with the proposed loss function performs best regarding errors, while the hinge loss function performs 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 when compared with cross-entropy loss. More generally, Fig. 4 and Fig. 5 show that the linear output layer with the improved loss function performs better than does a traditional output layer with cross-entropy loss and hinge loss.



(a)

(b)

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

4.3 Comparisons with state-of-the-art methods

Furthermore, we compare the classification performance of the proposed KBNN with those of state-of-the-art methods on three datasets.

4.3.1 Results on MNIST

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

Tab. 2 Classification results (%) on MNIST.

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

Tab. 3 Classification results (%) on CIFAR-10.

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

Tab. 4 Classification results (%) on CIFAR-100.

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

which was trained with an augmented training dataset by using distortion techniques. The test error results are summarized in Tab. 2.

As shown in Tab. 2, KBNN performs better on MNIST than most of the compared methods. In particular, KBNN achieves higher classification accuracy than the traditional combinational method DLSVM, which indicates that the kernel mapping contributes to the classification accuracy. The KBNN also outperforms the CNN+softmax, further showing the good performance of the proposed loss function. Compared with Niu's method, which uses distortion techniques to augment the training dataset and enhance generalizability, KBNN is trained directly on the original training set; hence, KBNN's results are weaker than those of Niu's method. However, the difference in classification accuracy is quite small: KBNN result trails those of Niu's method by 0.17.

4.3.2 Results on CIFAR-10

To further illustrate the generalizability of KBNN, we conducted comparisons with six image classification methods on the CIFAR-10 dataset. In this experiment, we compared KBNN with the combination method DLSVM, two improved loss function methods: large-margin Gaussian Mixture loss (ResNst110+L-GM [23]) and multi-loss function (ML-DNN [28]), and three high-performing methods: NIN [18], Maxout Networks [29] and Drop-Connect [30]. The test error results are presented in Tab. 3.

As Tab. 3 shows, KBNN performs the best on CIFAR-10. In particular, we do not need to enhance the network architecture of KBNN to address more complex datasets, which verifies that KBNN has a good generalization capability.

4.3.3 Results on CIFAR-100

To investigate the performance of KBNN on more complex dataset, we further compared KBNN with six represntive and well-performance methods on CIFAR-100 dataset: Learned Pooling [31], Stochastic Pooling [32], Maxout Networks, NIN, ML-DNN, and ResNet (110-layer) [33].

Tab. 4 gives the performance on the test error result of our proposed KBNN and other state-of-the-art methods. It can be seen that KBNN surpasses most methods with a test error of 32.71%. ResNet is an outstanding network in diverse applications, which uses deep residual learning to solve the problem of the difficulty to train a deeper network. Although the classification result of KBNN is lower than RestNet, KBNN has fewer network layers, which shows that KBNN can obtain better performances with relatively smaller model size. Generally, from the experimental results of Tab. 4, KBNN is also competitive in more complex dataset.

5 Conclusion and future work

In this study, we propose 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 establish 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 applied to 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. According to the experimental results on two commonly used datasets, 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, the upper and lower bounds of its value referred to the empirical settings in other literature. These empirical settings of parameters and architectures 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 National Natural Science Foundation of China (Grant Nos.61972227 and 61672018), in part by Natural Science Foundation of Shandong Province (Grant No.ZR2019MF051), in part by Primary Research and Development Plan of Shandong Province (Grant No.2018GGX101013), and in part by Fostering Project of Dominant Discipline and Talent Team of Shandong Province Higher Education Institutions

References

[1] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[2] E. BagarinaoTakio, T. Kurita, M. Higashikubo, and H. Inayoshi. Adapting svm image classifiers to changes in imaging conditions using incremental svm: An application to car detection. In *Proceedings of the Asian Conference on Computer Vision*, pages 363–372, 2009.

[3] Y. Guo, X. Jia, and D. Paull. Effective sequential classifier training for svm-based multitemporal remote sensing image classification. *IEEE Transactions on Image Processing*, 27(6):3036–3048, 2017.

[4] G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[5] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[6] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 11(4):541–551, 1989.

[7] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard. Multimodal deep learning for robust rgb-d object recognition. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 681–687, 2015.

[8] W. W. Shi, Y. H. Gong, X. Y. Tao, D. Cheng, and N. N. Zheng. Fine-grained image classification using modified dcnns trained by cascaded softmax and generalized large-margin losses. *IEEE Transactions on Neural Networks and Learning Systems*, 30(3):683–694, 2018.

[9] X. X. Niu and C. Y. Suen. A novel hybrid cnncsvm classifier for recognizing handwritten digits. *Pattern Recognition*, 45(4):1318–1325, 2012.

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

[11] D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, 195(1):215–243, 1968.

[12] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980.

[13] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Proceedings of European Conference on Computer Vision*, pages 818–833, 2014.

[14] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *Proceedings of International Conference on Learning Representations*, pages 1–16, 2014.

[15] F. L. Zhang, X. Wu, R. L. Li, J. Wang, Z. H. Zheng, and Hu S. M. Detecting and removing visual distractors for video aesthetic enhancement. *IEEE Transactions on Multimedia*, 20(8):1987–1999, 2018.

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

[17] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint*, arXiv: 1502.03167v3, 2015.

[18] M. Lin, Q. Chen, and S. Yan. Network in network. *arXiv preprint*, arXiv: 1312.4400, 2014.

[19] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314, 1989.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2011.

[21] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, technical report, University of Toronto, 2009.

[22] Y. Tang. Deep learning using support vector machines. *arXiv preprint*, arXiv: 1306.0239v4, 2015.

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

[24] H. Lee, R. Grosse, R. Rananth, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of Annual International Conference on Machine Learning*, pages 609–616, 2009.

[25] T. H. Chan, K. Jia, S. Gao, J. W. Lu, Z. N. Zeng, and Y. Ma. Pcanet: A simple deep learning baseline for image classification. *IEEE Transactions on Image Processing*, 24(12):5017–5032, 2015.

[26] E. Hosseini-Asl, J. M. Zurada, and O. Nasraoui. Deep learning of partbased representation of data using sparse autoencoders with nonnegativity constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 27(12):2486–2498, 2016.

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

[28] C. Xu, C. Lu, X. Liang, J. B. Gao, W. Zheng, T. J. Wang, and S. C. Yan. Multi-loss regularized deep neural network. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(12):2273–2283, 2016.

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

[30] L. Wan, M. Zeiler, S. Zhang, Y. LeCun, and R. Fergus. Regularization of neural networks using dropconnect. In *Proceedings of International Conference on Machine Learning*, pages 1058–1066, 2013.

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

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

[33] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Xinxin Liu received her B.E. degree from the School of Computer Science and Technology at 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 the Shandong University of Technology, Jinan, China, in 2000; his M.S. degree in applied mathematics from Shandong University, Jinan, China, in 2003; and his Ph.D. degree in computational geometry from Shandong University, Jinan, China, in 2007. He is now a professor at 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 the Qufu Normal University, Qufu, China, in 1994, and his Ph.D. degree from the Department of Mathematics of the Northwest University, Xian, China, in 1997. His current position is full professor in the Department of Mathematics, Shandong University, Jinan, China. His research interests include computer-aided geometric design and computation, computational geometry, and function approximation.

Kai Shao Kai Shao, received his B.E. degree from the School of Computer Science and Technology at Shandong University of Finance and Economics, Jinan, China, 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 Ziyi Sun, received her B.E. degree from the School of Computer Science and Technology at Shandong University of Finance and Economics, Jinan, China, 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 BS and ME in computer science from Shandong University in 1982 and 1984, respectively, and his Dr. Eng. degree in computer science from the Tokyo Institute of Technology, Japan, in 1994. From 1997 to 2000, Dr. Zhang held visiting position at the University of Kentucky, USA. His research interests include CAGD, CG, information visualization and medical image processing.

Response to Reviewers