Fine-grained Image Classification using Convolutional Neural Network and Support Vector Machine

Yu Shi[1], Tao Lin[1,*], Wei He[1], Biao Chen[1], Ruixia Wang[1], Nan Jiang[1], Yabo Zhang[1]
[1]School of Computer Science & Infomation Enineering, Shanghai Institute of Technology, Shanghai, China
*Corresponding author

Although vast neural network models can classify images, sub-classification for each image class requires retraining the exisiting model with new fine-grained data, subject to high training cost and uncertain classification accuracy. To solve the aforementioned problems, a method using Convolution Neural Network and Support Vector Machine is proposed, where the former extracts general features from images for sub-classification while the latter categorizes these features. The method can be rapidly deployed in most Internet of Things systems to identify various targets; its effectiveness in reducing training cost and improving classification accuracy is verified through experiments.

Index Terms—**Fine-grained image classification, feature extraction, convolutional neural network, support vector machine, face classification, IoT**

I. Introduction

FINE classification of images is to realize sub-class division of some basic classes [1]. Compared with coarse-grained images, fine-grained images usually share similar appearance and features [2]; posture, perspective, illumination, occlusion and background interference during data acquisition often create large inter-class differences and small inner-class differences, making classification more challenging [3].

A general model for image classification is *feature extraction + classifier*, which involves: image pre-processing, feature description and extraction, and classifier design. Usually, feature extraction and classifier design determine the performance of an algorithm.

Classification algorithms fall into two categories: traditional and modern ones [4]. With manual feature extraction, traditional algorithms such as Support Vector Machine(SVM) is appropriate for simple image classification with fast training speed and low training cost. For fine image classification: the difference among classes is small, the interference from similar images is significant, manually extracting sufficient fine-grained features is costly, making the traditional algorithms infeasible.

Modern classification algorithms are built upon deep learning and Convolutional Neural Network(CNN). The CNN automatically extracts image features and outperforms traditional feature extraction methods such as PCA [5]; deeper networks achieves better performance under dramatically growing training cost.

However, for the CNN model that has been trained for coarse-grained classification, it is necessary to retrain the model or even change the model structure if fine-grained classification is further carried out. The existing model cannot be reused. Specifically, for fine-grained image classification, the network for coarse classification must be adjusted, bringing extra training cost and uncertain accuracy. If using the CNN for coarse-grained classification to process the original data can achieve data dimensionality reduction, traditional classifiers such as SVM can be used directly for the data after dimensionality reduction without retraining a new neural network model. The reusability of the neural network model is improved and the training cost is reduced. Hence, this paper proposes a method combining CNN and SVM for fine-grained image classification. The CNN firstly extracts general features from images and reduce image dimension [6]; the SVM model are trained with these compressed data to further classify coarse-grained images. Notably, Support Vector multi-classification machine with one-rest classification strategy is adopted for multi-label image classification.

The usage of SVM in an artificial neural network architecture is gradually attracting the attention of researchers and there are already many CNN-SVM classifiers that have been successfully applied in many fields. [7] attempts to use the CNN-SVM model for image classification. [8] try to use CNN-SVM model for early breast cancer detection systems based on patient's imagery. CNN-SVM model has been used in classification of the lung sounds [9]. Also, such model could use for the Fine-Grained Classification of Green Tea [10]. The above papers demonstrate the effectiveness of the CNN-SVM model, but in these models, the CNN and SVM models are trained and classified as a whole. In the proposed model, the CNN part is an already trained model, and the CNN part can be used alone for coarse-grained classification. By combining different SVM models, fine-grained classification of different categories can be accomplished.

As an application, rapid detection of people without wearing masks is critical in epidemic prevention and control. The proposed algorithm can easily be deployed to Internet of Things(IoT) terminals for target recognition

Manuscript received May 20, 2022; revised July 25, 2022. Corresponding author: Tao Lin (email: lintao@sit.edu.cn).

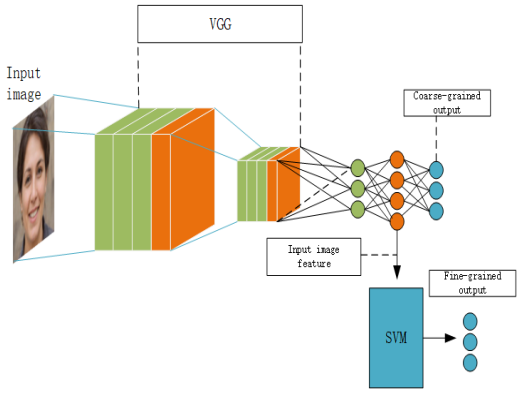


Fig. 1: CNN-SVM model.

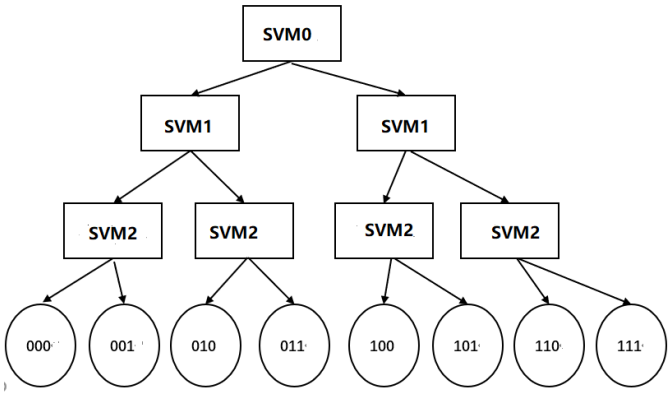


Fig. 2: An example of how images are divided into eight new sub-classes based on three different classification criteria through a three-tier SVM tree.

no longer needs to construct a larger neural network, but only needs to use the basic neural network combined with different SVM models.

II. CNN-SVM Model

The proposed model comprises two component stacks: a CNN and a SVM, as shown in Fig. 1. Dealing with identical images, the CNN accomplishes coarse-grained image classification while subsequently the SVM takes its output as extra image features to achieve fine-grained classification.

In short, this model utilizes both deep and shallow learning which extracts image features and classifies the extracted features respectively. If only the CNN is adopted, all image samples, requiring fine-grained classification or not, must be trained. In contrast, only samples really need fine-grained classification are considered by our model in the fine-classification stage, which greatly reduces training costs and achieves higher accuracy.

As illustrated in Fig. 1, the trained CNN model can reduce the dimension of training samples by extracting feature vectors. Particually, the dimension of the training samples are reduced from 12288 to 256 by the CNN; such low-dimensional feature vectors are then processed by the SVM to obtain the final classification result.

A. Multi-Label and Fine-Grained Classification

Multiple criteria should be used to accurately describe an image, creating multiple classification labels for each image, where various classification methods can be applied for each image class.

Since it is difficult to generate all classification labels simultaneously, in fine-grained image classification, image features can be prioritized in advance where lower priority features are considered only if higher ones fail to classify.

In the proposed model, SVM is used as a classifier to classify samples into multiple sub-classes according to different classification criteria. SVM is a binary classification model, with two well-known strategies to achieve multi-classification: one-one and one-rest [11], [12]. The one-one method builds a classifier for every pair of classes: requiring $\binom{n}{2}$ classifiers, and the class with most votes determines the classification result. The one-rest method constructs n classifiers for n classes, each distinguishes itself from the remaining classes; all classification results are integrated to determine the image class.

An SVM classification tree is adopted in this paper combining prioritized feature and one-rest strategy, to perform multi-label classification of images. The classification tree can effectively reduce the classification time, as the classification can be stopped immediately upon success.

In Fig. 2, a three-layer SVM tree is formed by three SVMs, each adopts the one-rest strategy to classify an image feature independently. These features are then combined to achieve fine-grained multi-label classification.

B. CNN-SVM Model Construction

The proposed CNN-SVM model shares the advantages of SVM and CNN. Compared with traditional feature extraction methods such as Principal Component Analysis(PCA), using CNN model can transform the original linearly inseparable features into linearly separable features, and quickly compress the size and dimensions of the samples. For the subsequent SVM model, using the feature information extracted by the CNN can accelerate training and improve prediction accuracy. Thus, we choose the CNN model as feature extractor.

Compared with other neural network models, the CNN model can better extract image feature information, reduce information loss, and swiftly reduce the dimension of images [13]–[16]. Numerous neural network models are generated based on CNN, such as LeNet, AlexNet, VGG and others. Among them, LeNet is one of the earliest CNN model with

60K parameters, whose structure can be described as "CONV1→CONV2→POOL2→FC3→FC4→ FC5→Softmax". Referring to other classic CNN models, our model consists of: a convolution layer, a pooling layer and a full connection layer. For building the CNN model, similar to the VGG model, a 3×3 convolution kernel is adopted. As LeNet and AlexNet, ReLu activation function is taken to minimize iteration and a dropout layer helps avoid over-fitting [17]–[19]. Following the last layer, a Softmax layer calculates the classification probability in the output layer, whose softmax function is shown in (1).

$$\begin{aligned} & softmax : \sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \\ s.t \quad & z = (z_1, \cdots z_K) \in \mathbb{R}^n \\ & i = 1 \cdots K \end{aligned} \tag{1}$$

The convolution kernel uses multiple 3×3 convolution kernels. After completing the convolution, we pad or trim the result to match subsequent network layers. Multiple methods are used to prevent over-fitting so that the model can better realize fine-grained image classification.

Shallow classification models based on statistical learning mainly includes: perceptron, K-nearest neighbor, Naive Bayes method, decision tree, SVM and Maximum Entropy model. For fine-grained image classification, SVM and decision tree (DTREE) are often used. The taller the decision tree, the higher its classification accuracy, with fewer fine-grained classes [20], [21].

SVM is a binary classification model, notable in linear and nonlinear classification, requiring a small number of samples for training. Consequently, it can be used as a classifier for fine-grained samples produced by the extracted features.

For linearly separable problems, SVM adopts maximum interval method, and the corresponding optimization problem can be expressed by:

$$\begin{aligned} & min_{w,b} \quad \frac{1}{2} \times \|w\|^2 \\ s.t \quad & y_i \times (w \times x_i + b) - 1 \geq 0 \\ & i = 1, 2 \cdots N \end{aligned} \tag{2}$$

The separated hyperplane obtained by maximizing the interval is:

$$w^* \times x + b^* = 0 \tag{3}$$

The corresponding SVM decision function is:

$$f(x) = sign(w^* \times x + b^*) \tag{4}$$

where the functional distance between the i_{th} sample point and the classified hyperplane is defined as:

$$\gamma_i = y_i \times (w \times x_i + b) \tag{5}$$

The i_{th}geometric interval is obtained by normalizing the function distance:

$$y_i(\frac{w}{\|w\|} \times x_i + \frac{b}{\|w\|}) \tag{6}$$

For approximately linearly separable training samples, a relaxation variable ε should be added, and the corresponding optimization problem becomes:

$$\begin{aligned} & \min_{w,b,\varepsilon} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{N} \varepsilon_i \times \frac{1}{2} \|w\|^2 \\ s.t \quad & y_i(w \times x_i + b) \geq 1 - \varepsilon_i \\ & i = 1, 2, \cdots N \end{aligned} \tag{7}$$

Nonlinear classification problems can be transformed into higher dimensional linear ones by nonlinear transformation, where kernel function is introduced to replace the inner product in nonlinear transformation.

$$K(x, z) = \phi(x) \times \phi(z) \tag{8}$$

The nonlinear SVM can be expressed as:

$$f(x) = sign(\sum_{i=1}^{N} \alpha_i^* \times y_i \times K(x, x_i) + b^*) \tag{9}$$

SVM uses SMO algorithm for fast learning. It keeps decomposing the original quadratic programming problem into quadratic programming subproblems of 2D variables and solving the subproblems, until all variables have met the KKT conditions.

Multiple SVMs can combine multiple class labels generated in fine-grained classification to generate new subclasses and obtain more accurate descriptions of images. Since SVM requires fewer training samples and less training time, the CNN-SVM model is adopted to achieve fine-grained image classification.

III. CNN Optimization and Training

First pre-train a CNN model that can be used for coarse-grained classification as the CNN part of the entire model. To further verify the effectiveness of our proposed method, we continue to train a CNN model that can be used for fine-grained classification based on the coarse-grained classification CNN model. In order to achieve better performance, this model is optimized in several aspects to improve the classification accuracy, and the training samples are enhanced to prevent overfitting.

A. Model Optimization

During training the CNN model, to accelerate and avoid local optimum, the momentum gradient descent method is adopted; a variable V is introduced, and the exponential decay method is used to record historical gradient data. If in current gradient direction, data is consistent with the historical gradient data, it is equivalent to increase the gradient descent in current direction ; otherwise, it decreases the descent. Note that gradient descent with momentum(GDM) can be expressed as:

$$GDM: \begin{cases} V_{dW} & = \beta V_{dW} + (1-\beta)dW \\ V_{db} & = \beta V_{db} + (1-\beta)db \\ W & = W - \alpha V_{dW}, b = b - \alpha V_{db} \end{cases} \tag{10}$$

The larger β is, the greater the influence of historical data on current gradient descent. The value of β is usually 0.5, 0.9, or 0.99. Due to some noise in the training samples, batch normalization is used to normalize data and PReLu function is used as the activation function to further accelerating model convergence and improving the robustness of the neural network model. PReLu can be expressed as :

$$PReLu(x) = \begin{cases} x & x > 0 \\ a \times x & x \leq 0 \end{cases} \tag{11}$$

Batch normalization(BN) focuses on a layer in the network, processing m training samples at a time, and standardizes the output of the j_{th} neuron in that layer:

$$BN: \begin{cases} \mu_j & = \frac{1}{m}\sum_{i=1}^{m} Z_j^i \\ \sigma_j^2 & = \frac{1}{m}\sum_{i=1}^{m} \frac{1}{m}\sum_{j=1}^{m}(Z_j^i - u_j)^2 \\ \hat{Z}_j & = \frac{Z_j - \mu_j}{\sqrt{\sigma_j^2 + \varepsilon}} \end{cases} \tag{12}$$

The data were normalized so that the mean value of the input eigenvalues of each layer is 0 and the variance is 1. To further improve the accuracy of the CNN, data enhancement is carried out on the training data. In the validation stage, the cross-validation method is adopted to avoid over-fitting.

B. Training Process

PubFig is a public face dataset of Columbia University, which contains more than 58k face images of 200 individuals, mainly used for face recognition in unrestricted scenes. It supplies face images with background information, range coordinates of face and basic image features. In this paper, the CNN model is trained using this data set. In data preprocessing, the training data are processed so that ordinary training samples can be applied to fine-grained classification training. We need to further classify the existing training samples to produce training samples with fine-grained class labels.

$$loss = \sum_{j=1}^{m}\sum_{i=1}^{n} -y_{ij}\log \hat{y_{ij}} - (1-y_{ij})\log(1-\hat{y_{ij}}) \tag{13}$$

We use cross entropy as the loss function. In the above formula, m is the sample size in batch, and n is the number of classes. To compute cross entropy, sample labels should be one-hot coded. In multi-classification, it is necessary to average the P_i, R_i values of all classes to get the overall value of P and R; Macro Average(MA) method is used to get the global P and R values, which is shown in (14):

$$MA: \begin{cases} P_{macro} & = \frac{1}{C}\sum_{i=1}^{C} P_i \\ R_{macro} & = \frac{1}{C}\sum_{i=1}^{C} R_i \end{cases} \tag{14}$$

P_i and R_i represent the P value and R value of class i; C represents the number of classes. The disturbance caused by unbalanced data distribution in the data set can be eliminated by macro average. In particular, we use Dropout layers in our original model to prevent model overfitting during the actual training process, as shown in Fig. 3.

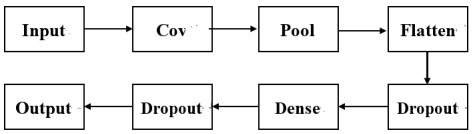
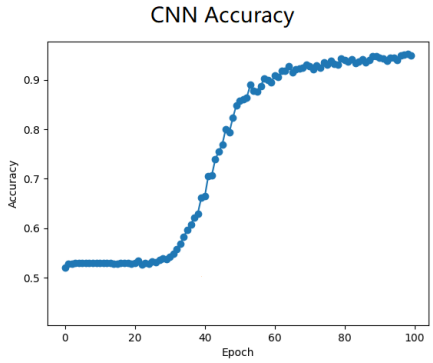


Fig. 3: Depicts the complete flow of a neural network with dropout layers.

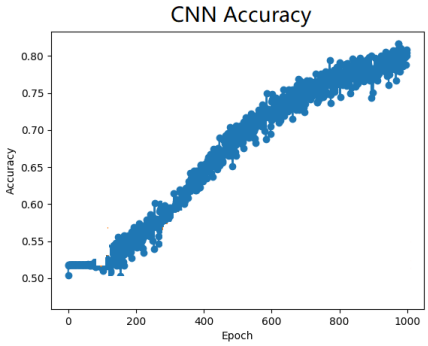
Fig. 4 to Fig. 5 plot the accuracy curve and lose curve for the training process. Fig. 4(a) and Fig. 5(a) demonstrats the results of the original model trained for 100 epochs without data enhancement. The training results show that the convergence speed of the model without data enhancement is very fast; it almost converges in 70 iterations. To improve the generalization ability of the CNN model and prevent over-fitting, we train the model with data enhancement: i.e. rotating the sample images in the training set randomly by an angle between 0 and 20 degrees, enlarging images randomly in horizontal and vertical direction, flipping the face images randomly in horizontal direction.

Fig. 4(b) and Fig. 5(b) shows the result after 1000 epoches of training with data enhancement. With data enhancement, both lose and accuracy curve, become smoother (See Fig. 4(a) and Fig. 5(a)), with slower regression speed and increasing fluctuation. After 1000 rounds of training, the training accuracy reaches more than 90%.

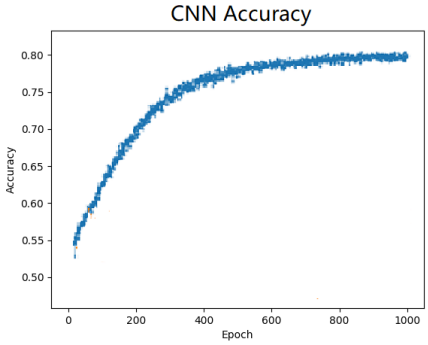
To accelerate model convergence and further reduce over-fitting to improve model accuracy, the batch normalization method is used to regularize the training data. Because batch normalization is not used simultaneously with the dropout layer, the latter are eliminated. The revised model is shown in Fig. 6 (BN stands for



(a) Original Accuracy Curve.

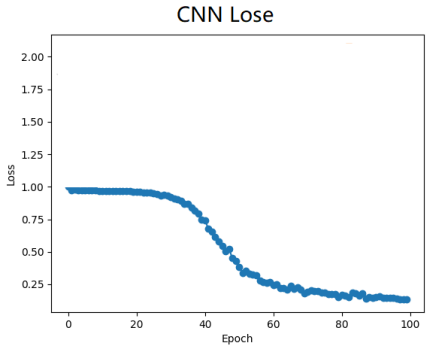


(b) Accuracy Curve after Data Enhancement.

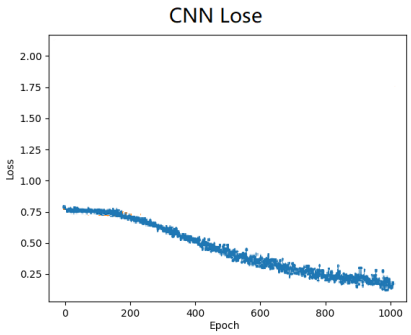


(c) Accuracy Curve after Batch Normalization.

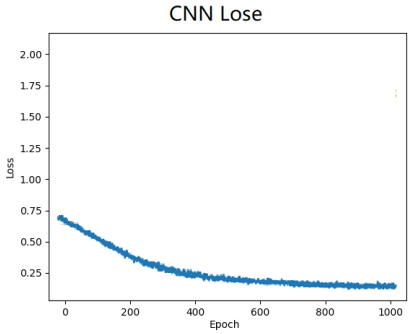
Fig. 4: Accuracy Curve during Training.



(a) Original Lose Curve.



(b) Lose Curve after Data Enhancement.



(c) Lose Curve after Batch Normalization.

Fig. 5: Lose Curve during Training

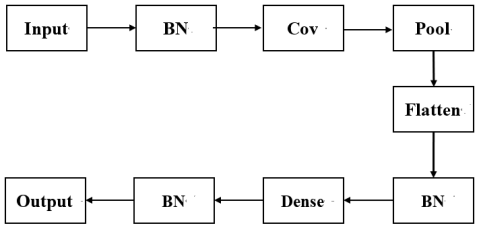


Fig. 6: Flow chart of CNN with batch normalization.

batch normalization layer). As batch normalization layer is inserted at the beginning, end and middle of the model to reduce over-fitting and accelerate the convergence.

Fig. 4(c) and Fig. 5(c) shows accuracy curve and lose curve of the model on the training set after adding the batch normalization layer with data enhancement after 1000 epoches. The accuracy curve becomes steeper

and the convergence is completed in about 400 rounds. The addition of batch normalization layer significantly accelerates the convergence and reduces jitter during training. In each training process, we use momentum method to speed up training. After the above training, the final CNN model is obtained.

IV. Evaluation and Results Analysis

To verify the effectiveness of the CNN-SVM model, we evaluate the model on the PubFig data set. This paper intends to classify the face images into: images with glasses, and images with beard as shown in Fig. 7. The model was built and trained by the Keras platform. The neural network of the specific model adopts an 8-layer CNN model similar to VGG model. We described the training process of the CNN model in previous section. The final model uses BN to prevent overfitting instead of dropout, and we use PReLu as the activation function for each layer. The kernel functions of SVM adopt RBF as show in Eq.15 , parameter C is set to 50, γ is set to 0.005. By randomly selecting data in the data set, the data set is divided into training set, test set in the ratio of 8:2.

$$RBF : K(x,y) = \exp^{-\gamma\|x-y\|}, \gamma > 0 \tag{15}$$

The CNN-SVM should be train and estimate as a entirety. Among them, CNN part is a pre-trained model, so the parameters of CNN part are not changed during process of SVM training. SVM can complete the training with a small number of samples [22]. In Fig. 8, 2000 samples are divided into 10 batches for SVM training. First, the trained CNN model was used to extract the eigenvalues of the training samples, and the extracted eigenvalues are used as the new training samples to train the SVM. Only a few samples are needed in the training process of SVM. It can be seen from Fig. 8 that training sample input of different batches has little influence on the score of SVM, with fluctuations within 8 percents. The training results of the initial input of 200 samples are not significantly different from those of the completion of 2000 samples, and the score of SVM remains stable without notable fluctuation.

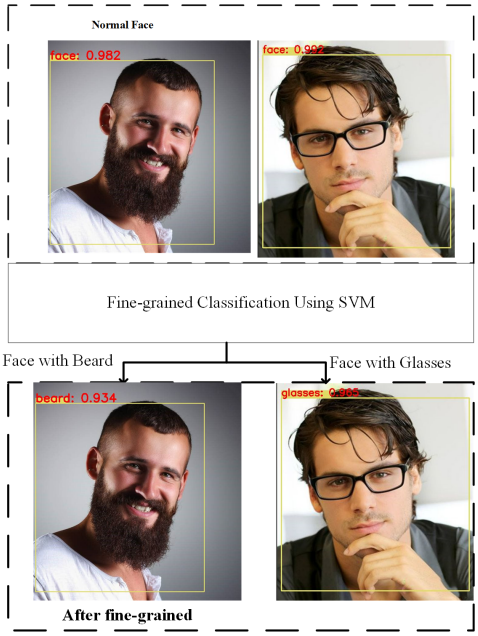


Fig. 7: Examples of Fine-grained face classification.

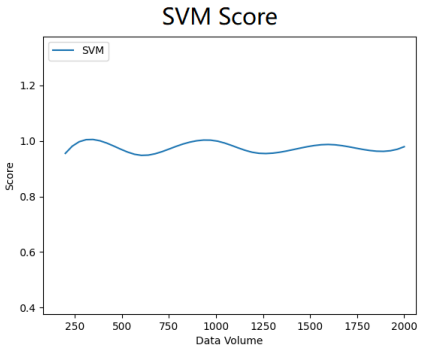
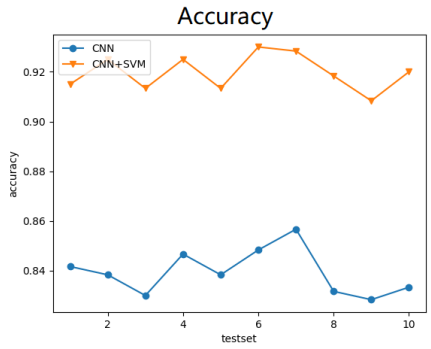


Fig. 8: With 2000 data input, SVM score transformation curve.

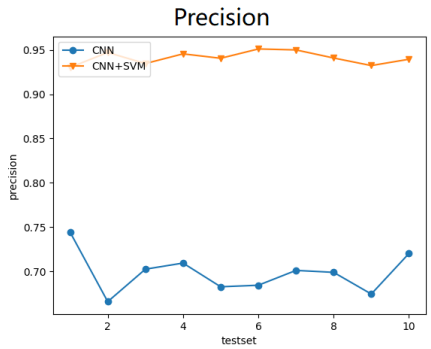
To further testify the performance of our model, we conduct experiments with several quantitative measures including Accuracy (A), Precision (P), and Recall (R); contrast methods are also selected to evaluate the effects of classification.

Fig. 9 shows the test results of the proposed *CNN-SVM* model and the CNN model further trained in the CNN part of the CNN-SVM model on 10 different test data sets. As can be seen in Fig. 9, the classification performance of the CNN-SVM model is far better than that of the CNN model alone. In fine-grained classification, the feature difference between data is minor, and the classification ability of CNN-SVM model is stronger than that of the CNN model alone. Therefore, the CNN-SVM model is suitable to deal with fine-grained target classification.

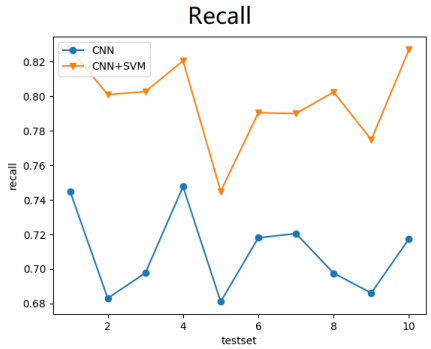
In fact, boosting model is common in image classification, which combines different models to enhance the classification ability. Most boosting models are divided into two parts: dimensionality reduction and classifica-



(a) Accuracy.



(b) Precision.



(c) Recall.

Fig. 9: Three different indicators were used: Accuracy (A), Precision (P) and Recall (R). Tests were carried out on 10 different data. As shown in 1, in different test sets, no matter which indicator CNN+SVM model had better classification performance than CNN model.

tion. PCA is one of the most important dimensionality reduction methods. It has a wide range of applications in the fields of data compression to remove redundancy and data noise removal. The main idea of PCA is to map n-dimensional features to k-dimensions. This k-dimension is a new orthogonal feature, also known as principal component, which is a k-dimension feature reconstructed on the base of the original n-dimensional feature. In fact, this is equivalent to retaining only the

TABLE I: CNN-SVM vs Other Classification Algorithms

Method	Accuracy(%)
PCA + LDA + Correlation	71
SVM	82.8
PCA + LDA + SVM	74.3
SVM + Adaboost	94.95
PCA + LDA + SVM + Adaboost	94.79
Simplified CNN model	75
CNN+SVM	96.5

dimension features containing most of the variance, and ignoring the feature dimensions containing almost 0 variance, so as to realize the dimensionality reduction processing of the data features. Therefore, many boosting models are constructed using the PCA method, such as PCA + LDA + Correlation, PCA+LDA+SVM, PCA + LDA + SVM + Adaboost. Most previous boosting models combines several shallow learning methods, whereas our proposed model combines deep and shallow methods as a new enhanced model. We demonstrate the effectiveness of our model by comparing it with the model constructed using the boosting models mentioned in [23]. The comparison results are shown in Table. I.

In Table. I, among the boosting models using the PCA method for dimensionality reduction, the classification accuracy of the model *PCA + LDA + SVM + Adaboost* is the highest of 94.79%. The CNN model in Table. I is a fine-grained classification model further trained on the basis of coarse-grained classification. Since the CNN model used in the experiment is a simplified VGG network model with shallow depth, the accuracy of fine-grained classification is only 75%. The accuracy of fine-grained classification using SVM alone is 82.8 %, accuracy of model *SVM + Adaboost* reaches 94.95%. The accuracy of fine-grained classification by adding SVM following the neural network has been greatly improved, reaching 96.5%, which proves the effectiveness of CNN-SVM model. Compared with other classification method, the accuracy of CNN-SVM model is higher. CNN-SVM inherits the advantages of CNN model in feature extraction and the advantages of SVM in feature classification. In terms of fine-grained classification, CNN-SVN model can extract sufficient fine-grained features and accurately classify them.

All combined models are designed following the extractor plus classifier pattern. The depth of the extracted features and the accuracy of the classifier will both have an impact on the final classification effect. In Table. I,the accuracy of using the shallow model as both the extractor and the classifier is lower than that of the CNN-SVM model, despite that the CNN model we use is a relatively simple neural network. The above results prove that we can obtain a fine-grained classification model with higher classification accuracy under a smaller training cost.

The proposed model can lead to many practical applications in the field of classification. For example, in the face detection task, we have already trained a CNN

model. But during the epidemic, we need to identify whether the detected face is wearing a mask upon detecting the face. At this time, if we continue to use the ordinary neural network, we need to recollect a large amount of data to retrain a new neural network, which is costly. By using the proposed method, it is only necessary to train the SVM for binary classification: whether masks are detected or not. Rather than train a new neural network, we can keep the overall network structrue and retrain the SVM. Note that the trainging cost for SVM is much lower than that of the whole model.

V. Conclusion and prospects

In order to achieve fine-grained image classification, the method *feature extractor + classifier* is adopted in this paper. The general features of a class can be proposed by coarse-grained image classification, and subsequently categorized by fine-grained classifier. We use the CNN model for feature extraction; experiments show that the underlying feature information can be better extracted than by traditional methods. Our proposed model can complete fine-grained classification without changing the structure of the existing neural network model.

Experiments indicate that low dimensional feature vector extracted from original data could be classified effectively, and the shallow learning model requires fewer training samples with faster training. This method greatly reduces the training cost and ensures the accuracy of final classification compared with retraining with a modified model. Due to the high training cost of using the VGG model directly, this paper adopts the simplified VGG neural network model and constructs a shallow CNN. Due to the shallow depth of the constructed neural network model, the precision of using the neural network directly for fine classification is low. However, after the feature vectors extracted from the network model are fed to the SVM model, the classification accuracy is improved by 23%. By comparing the classification accuracy with other combined models, we found the combined model using *deep learning + shallow learning* has better classification effect than other combinations of *feature extractor + classifier*. In the future, our model can be used in myriads of classification scenarios. For example, in the Internet of Vehicles systems, it can greatly improve the accuracy of target classification, demanding extremely low training costs.

Note that the neural network designed in this paper has a shallow depth and fails to effectively extract advanced features of images. Subtituting the CNN model for a deeper and more complex neural network model can further improve the performance of our model in fine-grained classification.

References

[1] D. Anguita and A. Boni, "Improved neural network for svm learning," *IEEE Trans Neural Netw*, vol. 13, no. 5, pp. 1243–1244, 2002.

[2] J. Tsiligaridis, "Classification with neural network and svm via decision tree algorithm," in *MATHEMATICAL METHODS AND COMPUTATIONAL TECHNIQUES IN SCIENCE AND ENGINEERING II*, 2018.

[3] T. M. Hamdani, A. M. Alimi, and M. A. Khabou, "An iterative method for deciding svm and single layer neural network structures," *Neural Processing Letters*, vol. 33, no. 2, pp. 171–186, 2011.

[4] G. Gao and M. Wüthrich, "Convolutional neural network classification of telematics car driving data," *Risks*, vol. 7, no. 1, 2019.

[5] F. Lei, X. Liu, Q. Dai, and W. K. Ling, "Shallow convolutional neural network for image classification," *SN Applied Sciences*, vol. 2, no. 1, 2020.

[6] A. Mahmood, M. Bennamoun, S. An, F. A. Sohel, F. Boussaid, R. Hovey, G. A. Kendrick, and R. B. Fisher, "Deep image representations for coral image classification," *IEEE Journal of Oceanic Engineering*, vol. 44, no. 1, pp. 121–131, 2019.

[7] Z. Ge, C. McCool, C. Sanderson, and P. Corke, "Modelling local deep convolutional neural network features to improve fine-grained image classification," in *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 4112–4116, IEEE, 2015.

[8] E. Deniz, A. Şengür, Z. Kadiroğlu, Y. Guo, V. Bajaj, and Ü. Budak, "Transfer learning based histopathologic image classification for breast cancer detection," *Health information science and systems*, vol. 6, no. 1, pp. 1–7, 2018.

[9] F. Demir, A. Sengur, and V. Bajaj, "Convolutional neural networks based efficient approach for classification of lung diseases," *Health information science and systems*, vol. 8, no. 1, pp. 1–8, 2020.

[10] D. Yu and Y. Gu, "A machine learning method for the fine-grained classification of green tea with geographical indication using a mos-based electronic nose," *Foods*, vol. 10, no. 4, p. 795, 2021.

[11] M. A. Ebrahimi, M. H. Khoshtaghaza, S. Minaei, and B. Jamshidi, "Vision-based pest detection based on svm classification method," *Computers and Electronics in Agriculture*, vol. 137, p. 52–58, 2017.

[12] Y. Xu, Z. Chen, X. Xiang, and J. Meng, "An energy-efficient parallel vlsi architecture for svm classification," *Ieice Electronics Express*, vol. 15, no. 7, pp. 20180099–20180099, 2018.

[13] K. Raju, N. Sandhya, and R. Mehra, "Supervised svm classification of rainfall datasets," *Indian Journal of ence and Technology*, vol. 10, no. 15, pp. 1–6, 2017.

[14] Hinton, E. G., Salakhutdinov, and R. R., "Reducing the dimensionality of data with neural networks.," *Science*, 2006.

[15] M. J. Cannon, A. M. Keller, C. A. Thurlow, A. Perez-Celis, and M. J. Wirthlin, "Improving the reliability of tmr with non-triplicated i/o on sram fpgas," *IEEE Transactions on Nuclear Science*, vol. PP, no. 99, pp. 1–1, 2019.

[16] K. T. Ahmed, Shahida, and M. A. Iqbal, "Content based image retrieval using image features information fusion," *Information Fusion*, 2018.

[17] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, "Going deeper in spiking neural networks: Vgg and residual architectures," *arXiv e-prints*, 2018.

[18] D. Laptev, N. Savinov, J. M. Buhmann, and M. Pollefeys, "Ti-pooling: transformation-invariant pooling for feature learning in convolutional neural networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[19] P. Oza and V. M. Patel, "One-class convolutional neural network," *IEEE Signal Processing Letters*, vol. 26, no. 2, pp. 277–281, 2019.

[20] E. Chen, X. Wu, C. Wang, and Y. Du, "Application of improved convolutional neural network in image classification," in *2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*, 2019.

[21] L. Windrim, A. Melkumyan, R. J. Murphy, A. Chlingaryan, and R. Ramakrishnan, "Pretraining for hyperspectral convolutional neural network classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. PP, no. 99, pp. 1–13, 2017.

[22] P. Latorre-Carmona, V. J. Traver, J. Sánchez, and E. Tajahuerce, "Online reconstruction-free single-pixel image classification," *Image & Vision Computing*, 2019.

[23] J. M. Haut, M. E. Paoletti, J. Plaza, A. Plaza, and J. Li, "Visual attention-driven hyperspectral image classification," *IEEE Transactions on Geoence and Remote Sensing*, vol. 57, no. 99, pp. 8065–8080, 2019.

Yu Shi Master of Electronic information science, Shanghai Institute of Technology University.Research Field: Artificial intelligence, Machine vision.

Tao Lin In 2004, he received his Ph.D. degree in Computer Application technology from Northeastern University. He previously worked as a senior engineer at Huawei. His research interests include: wireless terminal chip development, wireless communication and data communication research and development, embedded operating system, embedded Internet and data fusion.

Wei He Dr. Wei He received his B.Sc. degree in computer science from Fudan University and M.SE. degree from Peking University, in 2005 and 2008 respectively; and Ph.D. degree from School of Computer Science, University of Waterloo in 2013. He is now an assistant professor in the School of Computer Science and Information Engineering, Shanghai Institute of Technology, Shanghai, China. His current research interests include high performance wireless positioning, Internet of Things, survivable network design, and Cyber-physical systems.

Biao Chen Master of Electronic information science, Shanghai Institute of Technology University.Research Field: Artificial intelligence, Machine vision.

Ruixia Wang Master of Electronic information science, Shanghai Institute of Technology University.Research Field: Artificial intelligence, Machine vision.

Nan Jiang Master of Electronic information science, Shanghai Institute of Technology University.Research Field: Artificial intelligence, Machine vision.

Yabo Zhang Master of Electronic information science, Shanghai Institute of Technology University.Research Field: Artificial intelligence, Machine vision.