Computers, Materials & Continua CMC, vol.65, no.3, pp.2065-2077, 2020

Adaptive Binary Coding for Scene Classification Based on Convolutional Networks

Shuai Wang[1] and Xianyi Chen[2, *]

Abstract: With the rapid development of computer technology, millions of images are produced everyday by different sources. How to efficiently process these images and accurately discern the scene in them becomes an important but tough task. In this paper, we propose a novel supervised learning framework based on proposed adaptive binary coding for scene classification. Specifically, we first extract some high-level features of images under consideration based on available models trained on public datasets. Then, we further design a binary encoding method called one-hot encoding to make the feature representation more efficient. Benefiting from the proposed adaptive binary coding, our method is free of time to train or fine-tune the deep network and can effectively handle different applications. Experimental results on three public datasets, i.e., UIUC sports event dataset, MIT Indoor dataset, and UC Merced dataset in terms of three different classifiers, demonstrate that our method is superior to the state-of-the-art methods with large margins.

Keywords: Scene classification, convolutional neural network, one-hot encoding, supervised feature training.

1 Introduction

Scene classification is an important and challenging research topic in the field of computer vision. This technology, involving various cross-cutting areas such as pattern recognition, computer vision systems, signal processing, human-computer interaction, and privacy preserving [Luo, Qin, Xiang et al. (2020)], is essential for solving the problems of image retrieval [Xia, Lu, Qiu et al. (2019)] and image recognition [Sun and Ponce (2016); Zheng, Jiang and Xue (2012); Liu, Cong, Fan et al. (2017)]. For a given image, scene classification is to recognize the content and information this image contains to determine the scene to which it belongs [Yang and Newsam (2011); Lazebnik, Schmid and Ponce (2006); Zuo, Wang, Shuai et al. (2014)]. In recent years, with the emergence of new scene classification challenges, various scene classification techniques

[1] Department of Radiology and Biomedical Research Imaging Center, University of North Carolina at Chapel Hill, Chapel Hill, USA.

[2] Mathematics and Computer Science, University of North Carolina at Pembroke, Pembroke, USA.

[*]Corresponding Author: Xianyi Chen. Email: 0204622@163.com.

Received: 22 January 2020; Accepted: 07 August 2020.

have emerged. The existing scene classification techniques [Fan, Bo and Zhao (2015)] have gone through three stages in the process of development.

Early scene classification algorithms are mainly based on raw features [Singh, Girish and Ralescu (2017)]. Raw features can not only describe the texture of the image, but also reflect the deep structure information of the image if used properly. Therefore, many studies have been working on the researches on various image features for many years. SIFT (Scale-Invariant Feature Transform) [Tareen and Saleem (2018)] was proposed to describe the local features of an image. This feature is usually utilized for outdoor scene classification. GIST feature [Li, Cheng and Yu (2015)] was proposed to roughly extract the context information of images. This feature stimulates the human's vision and is easy to use. HOG (Histogram Oriented Gradients) [Cai, Zhu, Zeng et al. (2018)] was proposed to represent outline and edge information, which is suitable for globally stable scenes. The HOG-based models have the problems of feature point redundancy and low computational efficiency. Wu et al. [Wu and Rehg (2011)] proposed GENTRIST to solve these drawbacks. These raw features mentioned above are widely used for scene classification.

Raw features have good performance in simple scene classification [Cheriyadat (2014); Gong, Wang, Guo et al. (2014)]. However, these features usually have little semantic information, making them not perform well under the complex scene classification tasks. Thus, the focus of researches shifted to the understanding of the high-level semantics of the scenes. Li et al. [Li, Su and Li (2010)] proposed OB (Object Bank) features based on high-level semantics, identifying the image's label by multiple target detectors. Sadeghi et al. [Sadeghi and Tappen (2012)] proposed a simple and effective representation of images called LPR (Latent Pyramidal Regions), which has a good performance on all scene classification data sets. Junja et al. [Junja, Vdaldi, Jawahar et al. (2013)] proposed BOP (bag of parts) features, this method filters out the similar information contained in an image and retains the region with significant differences. It not only collects common targets in the scene but also captures the abstract features.

With the development of technology, the computing performance of computers continues to grow. Scene classification also has reached a new stage. LeCun et al. [LeCun, Bengio and Hinton (2015)] proposed a new concept called deep learning, the appearance of which made the automatic extraction and integration of features in the scene possible. A variety of learning frameworks of deep learning have been proposed and utilized, such as Caffe [Jia, Shelhamer, Donahue et al. (2014)], Theano [Bergstra, Breuleux, Bastien et al. (2010)], TensorFlow [Abadi, Agarwal, Barham et al. (2016)]; Doersch and Efros (2013); Wang, Wang, Bai et al. (2013)] so on. Due to the complexity of scene classification, the normal machine learning methods do not perform well on large data sets like ImageNet [Deng, Dong, Socher et al. (2009)] and require prior knowledge to process the data. While CNN (Convolutional Neural Networks) [Razavian, Azizpour, Sullivan et al. (2014); Zeng, Dai, Li et al. (2019)] can automatically optimize features based on the target dataset, making it a proper method for scene classification.

Although the existing methods have succeeded in the scene classification field, they still have certain disadvantages. Raw feature based algorithms require to manually design features, which is time-consuming and requires expertise [Zhang, Jin, Sun et al. (2020)]. Deep learning-based methods require retraining of the model, and the extracted deep

features are difficult to process. To solve these problems, we propose a simple solution by employing the binary encoding method to simplify the representation of the extracted deep features.

The rest of our paper is organized as follows. In Section 2, we present our algorithm; In Section 3, we do several experiments to verify the performance of the proposed method; Finally, in Section 4, we draw a brief conclusion.

2 Proposed algorithm

In this section, we give the details about our proposed algorithm for scene classification. First, we outline the overall architecture of the proposed method, as shown in Fig. 1, it is consists of three stages: Feature extraction, Binary encoding and Classifier training; then we introduce the Resnet that we employ for feature extraction; Next, we explain the proposed one-hot encoding representation; Finally, we summarize the process of image classification using the proposed feature representation.

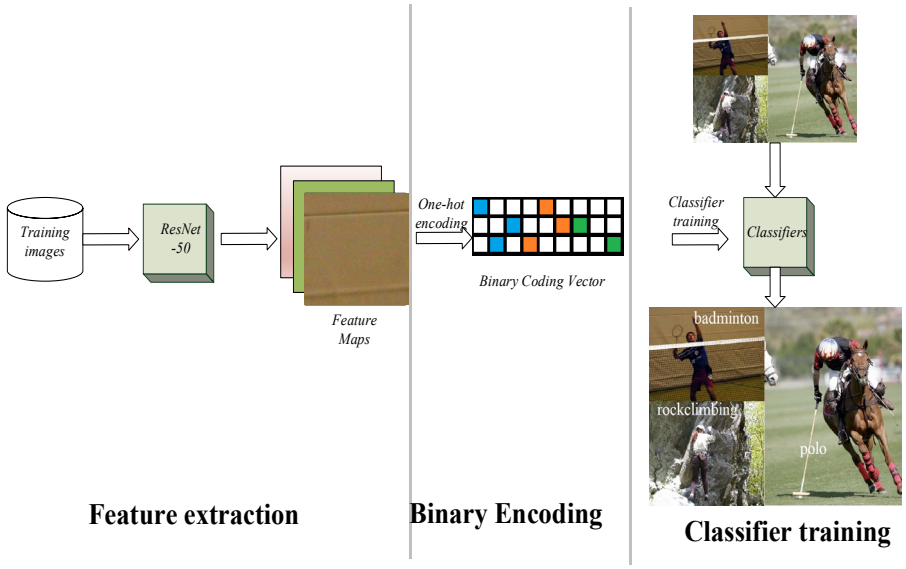

Figure 1: The framework of our proposed method, which consists of three stages: A) Feature extraction B) Binary encoding and C) Classifier training

2.1 Overall architecture

The first part of the algorithm is feature extraction as is shown in Fig. 2. In this step, the convolutional network ResNet is employed. Please note that any CNN-like architecture can be used here. In our implementation, Resnet-50 is used due to its excellent performance in image classification. For each training image in the database under consideration, we utilize the model trained on ImageNet to extract the deep features for further use.

Input: Image dataset $\boldsymbol{I}$ corresponding to intensity images, ResNet-50 model $\boldsymbol{M}$ trained on ImageNet, layer index L

Output: Deep Feature map $\boldsymbol{F}$

1) Initialize deep feature set $\boldsymbol{F} = []$
2) for $I_i \in \boldsymbol{I}$:
3) Input I_i to $\boldsymbol{M}$
4) Output the feature maps $F_i \in \boldsymbol{R}^{w \times h \times n}$ in layer L
5) Reshape $F_i \in \boldsymbol{R}^{w \times h \times n}$ to $F_i \in \boldsymbol{R}^{1 \times wh \times n}$
6) Update $\boldsymbol{F} = [\boldsymbol{F};\ F_i]$
7) end for

Figure 2: The pipeline of the feature extraction

Input: Deep Feature map $\boldsymbol{F}$, layer index L, number of clusters K

Output: Binary Feature map $\boldsymbol{B}$

1) Initialize binary coding feature set $\boldsymbol{B} = []$
2) for $n = 1, \ldots, N$ (N *is the channel number of feature maps in layer* L):
3) Initialize binary coding feature set for channel n as $B_n = []$
4) Cluster $F_{:,:,n}$ into K classes
5) for $I_i \in \boldsymbol{I}$:
6) Encode $F_{i,:,n}$ use one-hot binary coding vector B_{ni} based on the index of the nearest clustering center to $F_{i,:,n}$
7) Update $B_n = [B_n; B_{ni}]$
8) end for
9) Update $\mathrm{B} = [\mathrm{B}\ \mathrm{B_n}]$
10) end for

Figure 3: The pipeline of binary encoding

The second stage is binary encoding. The pseudo-code is shown in Fig. 3 below. In this part, we utilize one-hot binary encoding to turn the deep feature map into a binary feature map, making it possible to express the complex deep features with simple binary features which are suitable for classifier training.

The final step is the training part. The feature map then will be fed into the classifier along with the semantic labels of images to train the classifier. After training, the classifiers show good performance in terms of classification accuracy, which will be discussed in detail in the next section.

2.2 ResNet

ResNet is a convolutional neural network proposed by He et al. [He, Zhang, Ren et al. (2016)]. Deep convolutional networks naturally integrate the features of different levels, and deeper features can be extracted by deepening the hierarchy of the network. Thus, when building a convolutional network, the higher the number of layers is, the more features can be extracted via this network. However, when using deeper networks, gradient disappearance and explosion problems occur. This problem is largely solved by standard initialization and regularization layers, which ensure that networks with dozens of layers can converge, but with the increase in the number of layers, the gradient disappearance or the explosion problem still exists. Another problem is that network degradation. Suppose the designed structure of a network is deeper than the optimized structure of it, the redundant layers will cause network degradation.

The idea of ResNet is to assume that we design a network structure and there is an optimal network layer. Usually, the deep network we design has many redundant layers. Then we hope that these redundant layers can complete the identity mapping, ensuring that the input and output through the identity layer are identical. Which of the specific layers are the identity layers will be judged via the network training, ResNet changed the layers of the original network into a residual block. The concrete structure of the residual block is shown in Fig. 4.

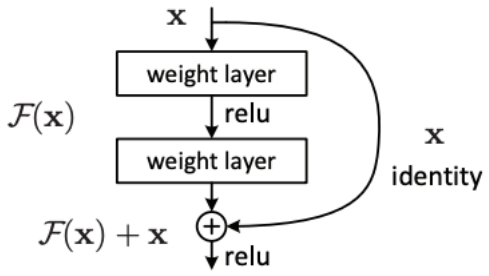

Figure 4: The residual block of ResNet [He, Zhang, Ren et al. (2016)]

ResNet avoids learning the parameters of the layer's identity map, using the structure shown above, let $h(x) = F(x) + x$; where $F(x)$ is called the residual term, we only need to learn $F(x) = 0$ to make this redundant layer map identically. Learning $F(x) = 0$ is easier than learning $h(x) = x$, because the parameter initialization in each layer of the network is generally biased towards 0.

ResNets have different types with different structures. In this paper, we employ the 50-layered ResNet-50, which is trained on the ImageNet Database, to extract the deep features of the images for classifier training. When extracting features, assuming the output of the model in layer L is a $w \times h \times n$ sized feature vector, we reshape its size to $1 \times wh \times n$, in order to put this vector into a feature map along with feature vectors extracted from other training images in the database. After this step, we will get a $g \times wh \times n$ sized feature map, where g represents the total amount of the training images.

2.3 One-hot encoding

In many machine learning tasks, features are not always continuous values, they may be categorical values. Thus, in the data preprocessing stage, non-numeric types are often quantized into numeric types to facilitate the input of the model. One-hot encoding is one of the methods to process the coding of the discrete data.

For each feature, if it has m possible values, after one-hot encoding the possible values will become m binary features. Moreover, these features are mutually exclusive, with only one activated at a time. Therefore, the data becomes sparse. For example, there are 3 features to describe a person, which are gender, nationality, and stature. Each feature has different values as is shown in Fig. 5 below.

Object: Person

Attribute: Gender, Nationality and Stature

$$Gender = \{male, female\}$$

$$Nationality = \{Chinese, Japanese, American, Korean\}$$

$$Stature = \{slim, normal, fat\}$$

Figure 5: The illustration of a person's attributes

If we want to describe a slim American woman, the binary feature after one-hot encoding is 010010100. More specifically, the feature can be divided into 3 parts: The first part is 01, taking up 2 bits which means $female$. The second part is 0010, taking up 4 bits, which means $American$; The last part is 100 meaning $slim$, taking up the bits left.

The one-hot encoding solves the problem that the classifier is not good at processing attribute data, Besides, it also helps expand features to a certain extent.

In our framework, for each channel of the feature map, we firstly use K-means to cluster the entire feature map into K classes. Then we encode the feature vectors of each image obtained in the feature extraction part into binary vectors based on the nearest clustering center. With the one-hot encoding method, we can turn the deep features extracted from the convolutional network into binary features to make the training process more effective and more efficient.

2.4 Image classification

The core of image classification is the task of assigning a label to an image from a given set of categories. In fact, this means that our task is to analyze an input image and return a label that categorizes the image, and those tags always come from a predefined set of possible categories. According to the different training conditions, the training method can be divided into supervised learning, unsupervised learning, and semi-supervised learning.

The neural network model and the encoding method mentioned above is a data pre-processing stage. To classify the image, we need to apply the pre-processed features to train the classifier.

To summarize, in this paper, we utilize a supervised learning method to train the classifiers. Given the database, feature vectors are extracted with the help of ResNet-50.

And we encode the vectors with the binary coding method to construct the feature maps. Due to the characteristics of the one-hot encoding, the binary coding method is adaptive. Therefore, we employ 3 different classifiers, which are: random forest, ensembles for boosting, and SVM. After training, we do the same procedure of feature extraction to the images to be classified and put them into the classifier. In other words, given the feature maps along with the semantic labels of the database, we use the 3 classifiers mentioned above to produce the classification results, which will be shown in the following section.

3 Experimental results

As we introduced in the previous part of this paper, the ResNet-50 is employed to extract the original deep features from the images. When doing the experiments, the ResNet-50 model we use is a pre-trained network downloaded from the internet. This model was trained on ImageNet. For each training image, we extract 2 feature vectors of different sizes. The first feature vector extracted is the output of the layer named 'res5c', which is a convolution layer of ResNet-50. The size of this feature vector is $7 \times 7 \times 2048$. The second feature vector extracted is the output of the layer named 'pool5', which is a pooling layer of ResNet-50. The size of this feature vector is $1 \times 1 \times 2048$. We use 3 image datasets in total and train 3 different classifiers for every dataset.

Table 1: Results on the UIUC dataset under different K

Value of K	Name of the classifier		
	Random Forest	Ensembles for Boosting	SVM
K=5	95.87 ± 0.70	95.23 ± 0.45	94.21 ± 0.55
K=10	94.95 ± 0.60	95.46 ± 0.49	93.29 ± 0.56
K=15	94.75 ± 0.49	94.63 ± 0.67	94.20 ± 0.59
K=20	93.93 ± 0.05	94.58 ± 0.50	93.10 ± 0.10

3.1 UIUC sports event dataset

UIUC sports event dataset contains 8 categories of sports: rowing (250 images), badminton (200 images), polo (182 images), bocce (137 images), snowboarding (190 images), croquet (236 images), sailing (190 images), and rock climbing (194 images).

First, we do some experiments to test the parameter K (K is the number of clusters we use at the encoding phase) to see if K has an influence on the performance of the proposed method. We choose different values of K (K=5, K=10, K=15, and K=20) to train the classifier with the UIUC dataset. For every K we do three times of tests. For every test, we choose 560 images in random from the dataset as the training set and 1014 images as the test set. The results are shown in Tab. 1.

We could see from Tab. 1 that with the increase of the parameter K, the accuracy of the proposed method only has minor changes in value. That means that K has little effect on the performance of the algorithm. However, with the increase of K, the size of the binary feature map extracted from the dataset grows in multiples, which makes the computing resources occupied by the algorithm also multiply. Besides, the algorithm already has a competitive performance when K=5 compared with another similar algorithm, which is shown in Tab. 2. The bold value in the table indicates the best results in the same column.

From Tab. 2, we could see that our method outperforms all the compared methods. Therefore, the experiments on the other 2 datasets are all tested under conditions of K=5.

Table 2: Comparison with other scene classification algorithms on the UIUC dataset

State-of-the-art Algorithms		Our method using different classifiers	
Methods	Accuracy	Methods	Accuracy
Hybrid-parts [Zheng, Jiang, and Xue (2012)]	84.5	Ours with Random Forest	$\mathbf{95.87 \pm 0.70}$
Hybrid-parts+GIST+SPM [Zheng, Jiang, and Xue (2012)]	86.3	Ours with Ensembles for Boosting	95.23 ± 0.45
MIDL [Wang, Wang, Bai et al. (2013)]	88.5 ± 2.3	Ours with SVM	94.21 ± 0.55
DeCAF [Donahue, Jia, Vinyals et al. (2014)]	**93.9**		
DSFL [Zuo, Wang, Shuai et al. (2014)]	86.5		
Discriminative Part Detect [Sun and Ponce (2016)]	86.8 ± 1.0		

3.2 MIT indoor dataset

This database contains 67 indoor categories with a total of 15620 images. The number of images varies across categories, but there are at least 100 images per category. When testing this database, we use 5460 images as the training set and 1340 images as the test set. The comparison with our algorithm in terms of accuracy is shown in Tab. 3 and the confusion matrices of 3 classifiers are also shown below as Fig. 6. In Fig. 6(a) represents the classification performance with Random Forest; (b) represents the classification performance with Ensembles for Boosting; (c) represents the classification performance with SVM. The vertical color bar indicates the proportions of samples over the actual class total. There are in total of 67 labels representing 67 indoor scenes in each matrix. The method with SVM produces the best classification performance.

From Tab. 3, we can see the proposed method also outperforms most of the state-of-the-art classification algorithms on the indoor scene recognition tasks. However, compared to DeepFeats_Mp [Gong, Wang, Guo et al. (2014)], which is the best among the compared algorithms, our best results are slightly lower. In the next experiment we test whether the algorithm is adaptive for remote sensing images with the help of another dataset.

Table 3: Comparison with other scene classification algorithms on the MIT Indoor dataset

State-of-the-art Algorithms		Our method using different classifiers	
Methods	Accuracy	Methods	Accuracy
Hybrid-parts [Zheng, Jiang and Xue (2012)]	39.8	Ours with Random Forest	57.69 ± 1.04
Mode Seeking [Doersch and Efros (2013)]	64.0	Ours with Ensembles for Boosting	57.61 ± 1.38
MIDL [Wang, Wang, Bai et al. (2013)]	50.2	Ours with SVM	$\mathbf{66.39 \pm 0.34}$

DeCAF [Donahue, Jia, Vinyals et al. (2014)]	58.5
DeepFeats_MP [Gong, Wang, Guo et al. (2014)]	**68.9**

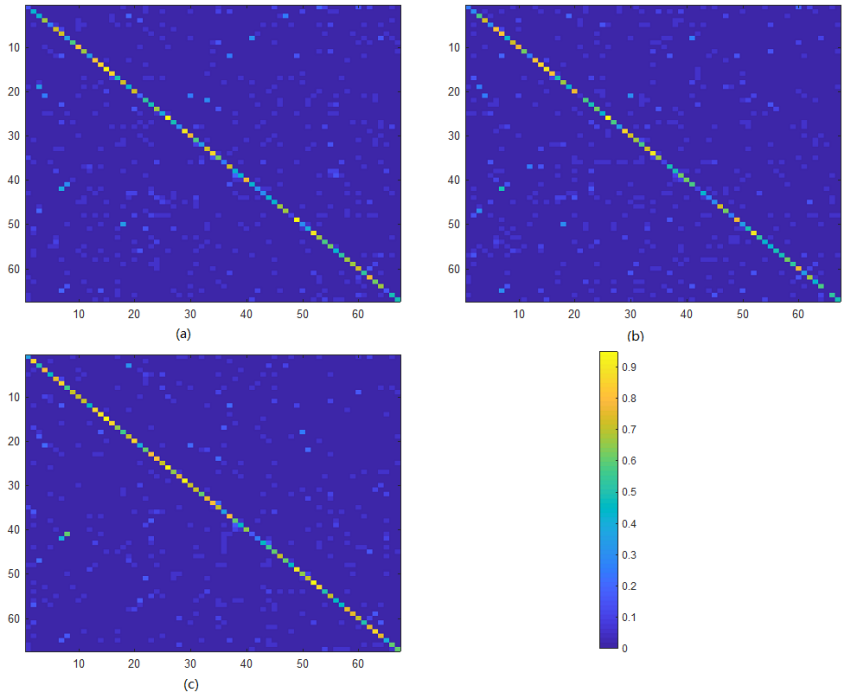

Figure 6: The confusion matrices of 3 classifiers on MIT indoor dataset

3.3 UC Merced land use dataset

The UC Merced dataset contains 21 categories of remote sensing satellite images manually extracted from large images of the USGS National Map Urban Area Imagery collection for various urban areas around the country. There are 100 images for each class. When testing this dataset, we randomly choose 1680 images as the training set and 420 images as the test set. The experiment is repeated for 3 times. And the comparison with our algorithm is shown in Tab. 4 together with the confusion matrices of 3 classifiers in Fig. 7. In Fig. 7(a) represents the classification performance with Random Forest; (b) represents the classification performance with Ensembles for Boosting; (c) represents the classification performance with SVM. The vertical color bar indicates the proportions of samples over the actual class total. There are in total of 21 labels representing 21 land scenes in each matrix.

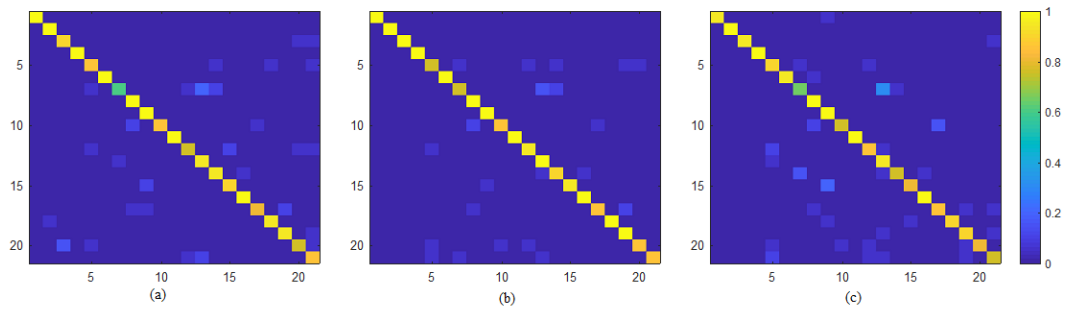

Figure 7: The confusion matrices of 3 classifiers on UC Merced dataset

In general, the experimental results prove that our method makes the training of the classifiers more effectively and is adaptive for various data sets. That is to say, our method can be applied to any convolutional networks. In addition, for a specific data set, employing our method with the help of suitable classifiers has a positive influence on the performance of classification in terms of accuracy.

Table 4: Comparison with other scene classification algorithms on the UC Merced dataset

State-of-the-art Algorithms		Our method using different classifiers	
Methods	Accuracy	Methods	Accuracy
SPMK [Lazebnik, Schmid and Ponce (2006)]	74.0	Ours with Random Forest	90.95 ± 1.08
SPCK++ [Yang and Newsam (2011)]	76.05	Ours with Ensembles for Boosting	$\mathbf{92.38 \pm 1.00}$
SIFT+SC [Cheriyadat (2014)]	81.67	Ours with SVM	88.33 ± 1.73
Saliency-guided [Fan, Bo and Zhao (2015)]	$\mathbf{82.72 \pm 1.18}$		

4 Conclusion

Compared to the traditional image classification methods based on convolutional networks, the proposed method takes an additional step of encoding the features before training the classifiers, making the training process more effective and more efficient. Unlike the previous studies focusing mostly on optimizing the structure of the neural networks, we propose a relatively uncomplicated way to improve the performance of the trained classifiers in this paper. After the feature extraction of training images with the help of ResNet-50, we encode these deep features into binary features first instead of directly using them for classifier training. By employing one-hot encoding, the original features, which are difficult to be understood by the classifiers because they are usually deep and complex, are turned into binary features, which are more suitable for classifier training. The experiments show the following:

1) By utilizing the framework proposed in this paper, even a basic classifier can be trained with the deep features extracted from the convolutional networks and obtain a relatively good performance.

2) The proposed algorithm can be utilized for training on different types of image datasets. Besides, the method is adaptive for various classifiers.

Funding Statement: This work is supported by the National Key R&D Program of China 2018YFB1003205; by the National Natural Science Foundation of China U1836208, U1536206, U1836110, 61972207; by the Engineering Research Center of Digital Forensics, Ministry of Education; by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund; by the Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET) fund, China.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Citro, C. et al. (2016): TensorFlow: large-scale machine learning on heterogeneous distributed systems. arXiv160304467 Cs. http://arxiv.org/abs/1603.04467.

Bergstra, J.; Breuleux, O.; Bastien, F.; Lamblin, P.; Pascanu, R. et al. (2010): Theano: a CPU and GPU math expression compiler. *Proceedings of the Python for Scientific Computing Conference*, vol. 4, no. 3.

Cai, L.; Zhu, J.; Zeng, H.; Chen, J.; Cai, C. et al. (2018): HOG-assisted deep feature learning for pedestrian gender recognition. *Journal of the Franklin Institute*, vol. 355, no. 4, pp. 1991-2008.

Cheriyadat, A. (2014): Unsupervised feature learning for aerial scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 1, pp. 439-451.

Deng, J.; Dong, W.; Socher, R.; Li, L. J.; Li, F. F. (2009): ImageNet: a large-scale hierarchical image database. *Proceedings of Computer Vision and Pattern Recognition*, pp. 248-255.

Doersch, C.; Efros, A. (2013): Mid-level visual elemssent discovery as discriminative mode seeking. *Advances in Neural Information Processing Systems*, vol. 1, pp. 494-502.

Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N. et al. (2014): Decaf: a deep convolutional activation feature for generic visual recognition. *International Conference on Machine Learning*, pp. 647-655.

Fan, Z.; Bo, D.; Liangpei, Z. (2015): Saliency-guided unsupervised feature learning for scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 4, pp. 2175-2184.

Gong, Y.; Wang, L.; Guo, R.; Lazebnik, S. (2014): Multi-scale order-less pooling of deep convolutional activation features. *European Conference on Computer Vision*, pp.184-199.

He, K.; Zhang, X.; Ren, S.; Sun, J. (2016): Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778.

Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J. et al. (2014): Caffe: convolutional architecture for fast feature embedding. *Proceedings of the ACM Conference on Multimedia*, pp. 675-678.

Juneja, M.; Vedaldi, A.; Jawahar, C.; Isserman, A. (2013): Blocks that shout: distinctive parts for scene classification. *Proceedings of Computer Vision and Pattern Recognition*, pp. 923-930.

Lazebnik, S.; Schmid, C.; Ponce, J. (2006): Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 2169-2178.

LeCun, Y.; Bengio, Y.; Hinton, G. (2015): Deep learning. *Nature*, vol. 521, no. 7553, pp. 436-444.

Li, B.; Cheng, K.; Yu, Z. (2015): Histogram of oriented gradient based gist feature for building recognition. *Computational intelligence and neuroscience*, vol. 127, no. 6, pp. 3489-3494.

Li, L.; Su, H.; Li, F. (2010): Object bank: a high-level image representation for scene classification & semantic feature sparsification. *Proceedings of Conference on Neural Information Processing Systems*, pp.1378-1386.

Liu, H., Cong, Y., Wang, S., Fan, H., Tian, D. et al. (2017): Deep learning of directional truncated signed distance function for robust 3D object recognition. *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Luo, Y. J.; Qin, J. H.; Xiang, X. Y.; Tan, Y.; Liu, Q. et al. (2020): Coverless real-time image information hiding based on image block matching and dense convolutional network. *Journal of Real-Time Image Processing*, vol. 17, no. 1, pp.125-135.

Razavian, A.; Azizpour, H.; Sullivan, J.; Carlsson, S. (2014): CNN features off-the-shelf: an astounding baseline for recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 806-813.

Sadeghi, F.; Tappen, M. (2012): Latent pyramidal regions for recognizing scenes. *Proceedings of European Conference on Computer Vision*, pp. 228-241.

Singh, V.; Girish, D.; Ralescu, A. (2017): Image understanding-a brief review of scene classification and recognition. *MAICS*, pp. 85-91.

Sun J.; Ponce, J. (2016): Learning dictionary of discriminative part detectors for image categorization and cosegmentation. *International Journal of Computer Vision*, vol. 120, no. 2, pp. 111-133.

Tareen, S. A. K.; Saleem, Z. (2018): A comparative analysis of sift, surf, kaze, akaze, orb, and brisk. *International Conference on Computing, Mathematics and Engineering Technologies*, pp. 1-10.

Wang, X.; Wang, B.; Bai, X.; Liu, W.; Tu, Z. (2013): Max-margin multiple-instance dictionary learning. *International Conference on Machine Learning*, pp. 1883-1891.

Wu, J.; Rehg, J. (2011): CENTRIST: a visual descriptor for scene categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1489-1501.

Xia, Z.; Lu, L.; Qiu, T.; Shim, H.; Chen, X. et al. (2019): A privacy-preserving image retrieval based on AC-coefficients and color histograms in cloud environment. *Computers, Materials & Continua*, vol. 58, no. 1, pp. 27-43.

Yang, Y.; Newsam, S. (2011): Spatial pyramid co-occurrence for image classification. In *International Conference on Computer Vision*, pp. 1465-1472.

Zeng, D.; Dai, Y.; Li, F.; Wang, J.; Sangaiah, A. K. (2019): Aspect based sentiment analysis by a linguistically regularized CNN with gated mechanism. *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 5, pp. 3971-3980.

Zhang, J.; Jin, X.; Sun, J.; Wang, J.; Sangaiah, A. K. (2020): Spatial and semantic convolutional features for robust visual object tracking. *Multimedia Tools and Applications*, vol. 79, no. 21, pp. 15095-15115.

Zheng, Y.; Jiang, Y. G.; Xue, X. (2012): Learning hybrid part filters for scene recognition. *European Conference on Computer Vision*, pp. 172-185

Zuo, Z.; Wang, G.; Shuai, B.; Zhao, L.; Yang, Q. et al. (2014): Learning discriminative and shareable Features for scene classification. *European Conference on Computer Vision*, pp. 552-568.