Research on Image Classification Method Based on Adaboost-DBN

Huadong Sun[1,2](✉), Wuchao Tao[1,2], Ran Wang[1,2], Cong Ren[1,2], and Zhijie Zhao[1,2]

[1] School of Computer and Information Engineering, Harbin University of Commerce, Harbin 150028, China
kof97_sun@163.com
[2] Heilongjiang Provincial Key Laboratory of Electronic Commerce and Information Processing, Harbin University of Commerce, Harbin 150028, China

Abstract. Image classification has been applied in many fields, which is an important branch of computer vision and pattern recognition. The boosting algorithm which is belong to ensemble learning can integrate several homogeneous classifiers, and combine the output layer's result of every classifier to improve the final classification accuracy. In this paper, the Adaboost-DBN algorithm is used to combine the four weak classifiers (DBN) and construct a strong classifier. The Adaboost-DBN algorithm is based on the Adaboost M1 algorithm and is used to achieve higher classification accuracy. The proposed algorithm is tested on the Corel-1K data set, and the result of classification is significantly improved comparing to other classifiers.

Keywords: Image classification · Adaboost-DBN

1 Introduction

Image classification is composed of feature extraction and feature classification. The most common feature of the image is the underlying visual features [1], including color features, texture features and shape features. Texture features are not based on pixel points, they need to be statistically calculated in regions containing multiple pixels. There are many methods to extract [2], such as the co-occurrence matrix, the local binary pattern (LBP), wavelet transform and so on. At present, texture features are widely used in image retrieval, face recognition and other fields. In this paper, the local neighborhood rotation right-angle pattern [3] is used to extract the texture features of the image. The method decomposes the image into R, G, B three-channel, and arranges them into spatial stereo in order, centering on the central pixel of the middle layer, and extracting features according to local neighborhood rotation. Finally, the feature value is dimension-reduced and expressed in the form of LBP.

Among the classification methods, the primitive method [4] has Naive Bayesian, K-Nearest Neighbor, Fisher Linear Discrimination and so on. With the rise of machine learning algorithms [5], classification algorithms such as Ensemble Learning, Convolutional Neural Networks, Deep Belief Networks, and Back Propagation Neural Networks have emerged, and the accuracy of classification of classifiers have been greatly

M. Jia et al. (Eds.): WiSATS 2019, LNICST 281, pp. 220–228, 2019.
https://doi.org/10.1007/978-3-030-19156-6_21

improved. Also it is suitable for large quantities and high levels of complexity. Many algorithms in machine learning are implemented in image classification. The Neural Networks can achieve non-linear fitting of images according to its complex network structure. Therefore, Neural Networks have been widely used. As a kind of boost algorithm for ensemble learning, Boosting algorithm combines several homogeneous classifiers into a strong classifier by linear weighting. The most widely used algorithm is the Adaptive Boost algorithm [6]. The Deep Belief Network (DBN) is composed of several restricted Boltzmann machines (RBM) and a layer of BP. The neural network contains many neurons and parameters (weights and offsets), so it has the strong ability of fitting. This article combines Adaboost and DBN to classify images. The traditional multi-classification method of boosting is based on the binary-class of "one-to-one" and "one-to-many" [7]. But the algorithm structure is cumbersome and the implementation of program wastes a lot of time. Therefore, this paper adopts the improved version of the Adaboost-DBN algorithm. It can output multi-classification directly and avoids complex computational processes. The algorithm is tested on the Corel-1K data set, and the classification accuracy show that it is greatly improved compared with the single classifier.

2 Feature Extraction

2.1 LBP

The LBP is a method for describing local texture features. The 3 * 3 local pixels are extracted to construct a local binary pattern, which is composed of a central pixel and a neighborhood pixel. LBP [3] uses a binary code in which each neighborhood pixel gray value is compared to the center pixel gray value to be converted into a binary coded form. The formula is as shown in (1) and (2):

$$LBP_{P,R} = \sum_{P=1}^{P} 2^{(P-1)} \cdot f_1(g_p - g_c) \tag{1}$$

$$f_1(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{2}$$

Where g_p is the gray value of the neighborhood pixel, g_c is the gray value of the center pixel, and P is the number of neighboring pixels. R is the radius of the neighborhood. Figure 1 shows the working process of LBP.

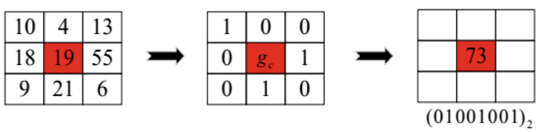

Fig. 1. LBP pattern.

2.2 VLBP

VLBP [8] is based on the dynamic texture change of adjacent frames in the video sequence. The dynamic texture of the image is calculated according to the gray value of the central pixel $g_{t_c,c}$ and the gray value of it's neighborhood pixel $g_{t,p}(t = t - L, t_c, t_c + L; P = 0, \cdots p - 1)$, p represents the number of neighboring pixels of the image center pixel. The formula can be expressed as:

$$VLBP_{L,P,R} = \sum_{q=0}^{3P+1} v_p 2^q. \tag{3}$$

The Fig. 2 shows the process of collecting samples of VLBP:

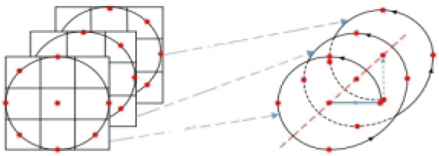

Fig. 2. The process of collecting samples of VLBP.

According to the collected samples obtained by VLBP (as shown in the Fig. 1), the central pixel is used as the reference point, and the local pattern values in the five directions are extracted showing in the Fig. 3:

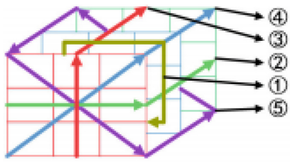

Fig. 3. The local pattern values in the five directions.

Feature extraction is the expression of images in a abstract fashion, which is the basis of image processing. Texture features characterize image information by analyzing the distribution of central pixels and their neighboring pixels. There are many methods for extracting texture features. In this paper, local neighborhood rotation right-angle patterns [3] are used to extract texture features. The algorithm steps are as follows:

(1) The original color image is pre-processed to separate its into R, G, B three-channel color;
(2) The single channel color is transformed by Haar wavelet, and the low frequency band is extracted as contour information;

(3) Based on the VLBP pattern, the three low frequency sub-bands are arranged in the order of R, G, B;
(4) The local neighborhood rotation right-angle patterns are used to extract feature extraction and it is represented into histogram form;
(5) The LBP pattern is used to reduce the dimension.

3 Adaboost-DBN Algorithm

3.1 Deep Brief Network

Belief in the Deep Belief Network refers to probability, so Deep Belief Network refers to a kind of neural network that can learn by probability. DBN is a probability generation model that aims to establish a joint distribution between data and labels. The DBN is composed of multiple restricted Boltzmann machine and an output layer. The data is putted into the neural network from the input layer. First, unsupervised training is performed by the restricted Boltzmann machine, and then supervised training is performed by the BP algorithm. The DBN can exploit the nonlinear relationship fully between the input data and labels, and finally get the weight and offset in the neural network.

Restricted Boltzmann Machine. The restricted Boltzmann machine is generated stochastic neural network proposed by Hinton and Sejnowski in 1986. The RBM model is consisted of a visible layer (v) and a hidden layer (h). The neurons between the layers are connected to each other, and the neurons in the layer are not connected. The restricted Boltzmann machine is used to perform unsupervised training between layer-by-layer on the neural network. The parameters obtained by training are used as the initial parameters of the neurons in each layer of the neural network, which are in a better location for space. RBM [9] is an energy-based model whose energy of the joint distribution of the visible variable (v) and the hidden variable (h) is:

$$E(\mathrm{v},\mathrm{h};\theta) = -\sum_{i,j} W_{i,j} v_i h_j - \sum_i b_i v_i - \sum_j a_j b_j \tag{4}$$

In (4) θ is the parameter $\{w, a, b\}$ of RBM, w is the weight of between the visible unit and the hidden unit, b and a is the bias of the visible unit and the hidden unit. With the energy of the joint distribution of v and h in (1), we can get the joint probability of v and h:

$$P_\theta(v, h) = \frac{1}{\mathrm{Z}(\theta)} \exp\left(\sum_{i=1}^{D} \sum_{j=1}^{F} W_{ij} v_i h_j + \sum_{i=1}^{D} v_i b_j + \sum_{j=1}^{F} h_j a_j \right) \tag{5}$$

Where Z(θ) is a normalization factor, also known as a partition function in (5), and $p_\theta(v,h)$ is find to get the edge distribution of h to get $p_\theta(v)$, and $p_\theta(v)$ is processed to get the parameters θ^* of RBM:

$$\theta^* = \arg\max L(\theta) = \arg\max \sum_{t=1}^{T} \log\ p(v^t|\theta) \tag{6}$$

After the parameters are acquired, they are updated according to the contrastive divergence algorithm proposed by Hinton. The expression is as follows:

$$\begin{aligned} \Delta\omega_{ij} &= \varepsilon(<v_i h_j>_{data} - <v_i h_j>_{recon}) \\ \Delta a_i &= \varepsilon(<v_i>_{data} - <v_i>_{recon}) \\ \Delta b_i &= \varepsilon(<h_j>_{data} - <h_j>_{recon}) \end{aligned} \tag{7}$$

Where ε is the learning rate of pre-training, $< >_{data}$ is the mathematical expectation of the training data set, and $< >_{recon}$ is the reconstructed mathematical expectation.

BP neural Network. BP neural network is a multi-layer feed forward neural network. Its main characteristic is: the signal is forward propagating, and the error is back propagating. In this thesis, The neural network is trained by BP algorithm, and the parameters of the network will eventually converge in a good position. The process of BP neural network is divided into two phases. The first phase is the forward propagation of the signal, the signal from the input layer through the hidden layer, and finally get to the output layer. The second phase is the back propagation of the error, there is an error between the output layer and the hidden layer, from the output layer and finally get to the input layer, which adjusts the weight and offset of the output layer to the hidden layer, and the weight and offset of the hidden layer to the input layer.

The BP algorithm [10] updates the parameters as follows, assuming that the number of nodes of the input layer is n, the number of nodes of the hidden layer is l, the number of nodes of the output layer is m, and the weight of the input layer to the hidden layer is ω_{ij}, implied the layer-to-output layer weight is ω_{jk}, the offset between input layer and the hidden layer is a_j, the offset between hidden layer and the output layer is b_k, and H_j is the output of the hidden layer. The learning rate is η, the excitation function is $g(x)$. The error between the expected output and the actual output is:

$$E = \frac{1}{2}\sum_{k=1}^{m}(Y_K - O_K)^2 \tag{8}$$

In (8) Y_k is the expected output and O_k is the actual output, as $Y_k - O_K = e_k$. An update formula for the weights and offsets obtained by E-derivation using the gradient descent method:

$$
\begin{aligned}
&\omega_{ij} = \omega_{ij} + \eta H_j(1 - H)x_i \sum_{k=1}^{m} \omega_{jk} e_k \\
&\omega_{jk} = \omega_{jk} + \eta H_j e_k \\
&a_j = a_j + \eta H_j(1 - H_j) \sum_{k=1}^{m} \omega_{jk} e_k \\
&b_k = b_k + \eta e_k
\end{aligned} \tag{9}
$$

3.2 The Proposed Adaboost-DBN Algorithm

The Adaboost algorithm is the most widely used in the boosting algorithm, and its central idea has not changed. That is, some weak classifiers are trained, the weight of the wrong samples in each training will become larger in the next training, and the weight of the correct samples will be reduced. The traditional algorithm is based on the two-category "one-to-one" and "one-to-many", dealing with multiple classifications. Because two methods have great redundancy in algorithm design and program implementation, this thesis uses the Adaboost M1algorithm [11] and made some improvements on this basis. The flow chart of the improved algorithm is:

(1) Initialize the weight distribution of the sample: $\mathrm{D1} = (\frac{1}{N}, \frac{1}{N}, \cdots, \frac{1}{N})$;
(2) $t = 1, 2, 3 \cdots T$;
  (1) Training the weak classifier DBN using the initialized weight samples;
  (2) The error of classifier: $\varepsilon_t = \sum_{i=1}^{m} D_i^t[h_t(x_i) \neq y_i], h_t(x_i)$ is the final value of the output layer, and its value is composed of 0 and 1, and is the same as $h_t(x_i)$ in step (4);
  (3) Weight of the classifier: $\alpha = \frac{1}{2}\lg\frac{1-\varepsilon^{(t)}}{\varepsilon^{(t)}} + \lg(K - 1)$;
  (4) The updated weight of sample: $\mathrm{D_i^{t+i}} = \mathrm{D_i^t} \cdot \exp(\alpha_t \cdot [h_t(x_i) \neq y_i])$;
  (5) Normalized the weight of $\mathrm{D_i^{t+i}}$;
(3) Obtained the strong classifier: $H(x) = \sum_{t=1}^{T} \alpha(t)h'(x), h'(x)$ represents the proportion of each sample in different class before the output layer output classification results, usually a decimal.

The Proposed Algorithm Analysis. There are two improvements between the proposed algorithm and the Adaboost M1 algorithm, the first improvement is step (3) the weight of the classifier, it is added a positive term $\lg(K - 1)$ [12] on the basis of the original. The term brings a great improvement to the performance of the algorithm. The original Adaboost M1 algorithm is asked that the correct rate of the classifier is greater than 1/2. It is difficult for the general classifier to meet this requirement. It is known from the improved classifier that the error of classifier meets $\varepsilon(t) < 1 - 1/K$. In other words, as the number of iterations increases, the accuracy of the classifier is reduced. The second improvement is step (6). The original is $H(x) = \arg\max_k \sum_{t=1}^{T} \alpha^{(t)}[h(x) = k]$, in the formula: $[h(x) = k]$ can be interpreted as, when the logical expression is true, then $[h(x) = k] = 1$, otherwise $[h(x) = k] = 0$, therefore the output form of $[h(x) = k]$ is a matrix consisting of 0 and 1. 1 means the result of classification is correct, 0 means the result of classification is error (the representation is the final value of the DBN output), then the α weights the value of the output layer. The improvement in this paper is $H(x) = \sum_{t=1}^{T} \alpha(t)h(x)$. The biggest difference compared with the original one is that $h(x)$ replaces the original $[h(x) = k]$, and $h(x)$ represents the proportion of each sample in the various types, usually it's a decimal. The $h(x)$ is weighted by the classifier weights, which can reflects the essential characteristics of each sample. The test has shown that improved algorithms can achieve better accuracy.

4 Experimental Results and Analysis

This experiment is compiled in the Matlab R2016b, using Corel-1K data set including 10 categories of 1000 pictures, which are divided into training set and testing set. In the proposed algorithm, the integrated weak classifiers are consisted of 4 DBNs, the number of input neurons and output neurons of are both 580 and 10, and the number of hidden layers are as: [100 100], [110 100], [90 90], [140 140]. The classifiers are Adaboost M1, DBN, and the proposed algorithm.

In order to fully exploit the classification ability of the classifier, the experimental data set is divided into 9 groups. The number of training set and testing set are (100, 900), (200, 800)…(900, 100). The classification result of the classifier is shown in the Fig. 4, in which the classification accuracy of the DBN takes the maximum value of the four results in each training (that is, the accuracy of DBN shown in the figure does not refer to a certain DBN). It can be seen from the figure that the classification curve of the classifier DBN coincides with the classification curve of Adaboost M1, which shows that the advantage of integrated learning that it can integrate the classification capabilities of classifiers to achieve higher performance. In the comparison of the data of the first to third groups and the ninth group, the classification ability of the Adaboost M1 algorithm is superior to the proposed algorithm in this paper. In the third to eighth groups, the proposed algorithm has the highest correct rate and reaches the maximum value (81.6%) in the test. The three classifiers achieved the best performance in the data (500, 500), which shown that the DBN has the best fitting capacity at this time. In the first four data sets, due to there are fewer training sets, the under-fitting of neural network leads to poor classification. Similarly, in the last four data sets, due to there are many training sets, the over-fitting of neural network also leads to poor results of classification.

The following table shows the classification accuracy of the three classifiers in the 9 data sets (%):

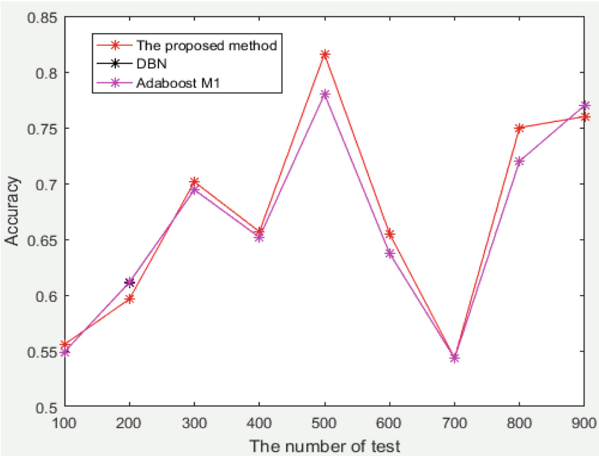

Fig. 4. The accuracy of three classifiers.

Table 1. The accuracy of three classifiers.

Train data	Test data	The proposed algorithm	DBN	Adaboost M1
100	900	0.556	0.5489	0.5489
200	800	0.5963	0.6112	0.6113
300	700	0.7014	0.6943	0.6943
400	600	0.6567	0.6517	0.6517
500	500	0.816	0.78	0.78
600	400	0.655	0.6375	0.6375
700	300	0.5433	0.5433	0.5433
800	200	0.75	0.72	0.72
900	100	0.76	0.77	0.77

It can be seen from the Table 1. that the accuracy of the proposed algorithm is significantly improved compared with the traditional classification. There are two reasons for this. Firstly, ensemble learning can integrate multiple classifiers, which fully exploits the classification capacity of each classifier; Furthermore, the traditional ensemble learning is improved which can get better performance.

5 Conclusion

Adaboost algorithm belongs to the category of decision-level fusion, it can superimposes the classification results of several weak classifiers to obtain a stronger classification effect. In this paper, the texture features of the image are extracted by the local neighborhood rotation right-angle pattern, and then these features are classified by DBN. The training process of DBN is divided into two steps: First, the RBM training process makes the weight and offset inside the neural network in a good position. Then, the BP algorithm feeds back the output layer error and further adjusts the value of the neural network parameter. The proposed algorithm in this paper is improved on the basis of Adaboost M1, the advantage is that it not only expands the selection range of the weak classifier, but also makes a deep excavation of the image features to make the classification accuracy significantly improved.

Acknowledgement. The Humanities and social sciences research projects of the Ministry of Education (18YJAZH128) and the Basic Scientific Research Operating Expense Project of Provincial Institutions of Higher Education in Heilongjiang (17XN003).

References

1. Patil, J.K., Kumar, R.: Analysis of content based image retrieval for plant leaf diseases using color, shape and texture features. Eng. Agric. Environ. Food **10**, 69–78 (2016)
2. Srivastava, P., Khare, A.: Integration of wavelet transform, local binary patterns and moments for content-based image retrieval. J. Vis. Commun. Image Represent. **42**, 78–103 (2017)
3. Hong, T.: Image retrieval technology research based on local neighborhood rotation right-angle pattern. Harbin University of Commerce (2017)
4. Cui, B., Ma, X., Xie, X., Ren, G., Ma, Y.: Classification of visible and infrared hyperspectral images based on image segmentation and edge-preserving filtering. Infrared Phys. Technol. **81**, 79–88 (2017)
5. Al-Mudhafar, W.J.: Integrating well log interpretations for lithofacies classification and permeability modeling through advanced machine learning algorithms. J. Petrol. Explor. Prod. Technol. **7**(4), 1023–1033 (2017)
6. Hassan, A.R., Bhuiyan, M.I.H.: An automated method for sleep staging from EEG signals using normal inverse Gaussian parameters and adaptive boosting. Neurocomputing **219**, 76–87 (2017)
7. Lv, Y., Hou, Z., Zhang, K.: Study of multi-class BP-AdaBoost and its application. High Technol. Lett. **25**(05), 437–444 (2015)
8. Kong, J., Zhan, Y., Chen, Y.: Expression recognition based on VLBP and optical flow mixed features. In: Fifth International Conference on Image and Graphics, ICIG (2009)
9. Probst, M., Rothlauf, F., Grahl, J.: Scalability of using restricted Boltzmann machines for combinatorial optimization. Eur. J. Oper. Res. **256**(2), 368–383 (2017)
10. Ding, W., Zhang, J., Leung, Y.: Prediction of air pollutant concentration based on sparse response back-propagation training feedforward neural networks. Environ. Sci. Pollut. Res. **23**(19), 19481–19494 (2016)
11. Zhang, Z.: The improvement and application of Adaboostalgorithm. Lanzhou University (2017)
12. Hu, J., Luo, G., Li, Y., Wang, C., Yu, X.: An AdaBoost algorithm for multi-class classification based on exponential loss function and its application. Acta Aeronaut. ET Astronaut. Sin. (04), 811–816 (2008)