A Vehicle Type Recognition Method based on Sparse Auto Encoder

H.L. Rong, Y.X. Xia
Wu Han University of Technology
WuHan, China

***Abstract*—In recent years, feature learning methods based on unsupervised learning get more and more attention. Until now, Unsupervised feature learning has been applied to solve many problems such as detection, recognition and classification. In this paper, we propose a deep feature learning method based on Sparse AutoEncoder to recognize vehicle types and to improve the classification accuracy rate. First we used Sparse AutoEncoder to generate the convolutional kernel and used the convolutional kernel to generate convolutional feature. Then pooling was applied. We repeated the network several times to construct a deep feature learning framework. To improve performance, we also combined the feature learned in different layer to form a new feature vector and applied PCA to reduce the dimension. Finally we used softmax to recognize the vehicle type. Adopting the local receive field, we can reduce the parameters. The experiment shows that our method can improve the performance a little.**

Keywords-vehicle type recognition; sparse AutoEncoder; multi-level features fusing

I. INTRODUCTION

Since the study of deep learning made great progress in 2006, feature learning methods based on neural network get more and more attention. Unsupervised feature learning[4-5] is a method that can learn feature from image automatically. The main advantage of unsupervised feature learning is that the feature learned by unsupervised feature learning is good for classification. AutoEncoder is a famous unsupervised feature learning method. AutoEncoder usually consists of two phases, the encoding procedure and the decoding procedure. AutoEncoder is trained by making the distance between the input and the output as small as possible. So AutoEncoder can learn the major feature of the input.

Recently many researches have been done about unsupervised feature learning based on AutoEncoder. Hai T. Phan[6] use linear regression feature in image preprocessing to divide the image to different sets, then different Stacked Sparse AutoEncoder is used to learn high level feature from different image set. Finally softmax and svm are used to classify the image. They archive high accuracy. Ragheb Walid[7] use Sparse Deep Belief Network and Denoising AutoEncoder to learn image representation. Ming Zhun[8] also learn image feature by Stacked AutoEncoder and use Deep Belief Network to perform classification which works well. Although Sparse AutoEncoder has been used to solve many problems,it is rarely used in vehicle type recognition problem.

In this paper, we proposed a novel vehicle type recognition framework based on Sparse AutoEncoder and Convolutional Network[9]. In our framework, we combined local receive field and Sparse AotoEncoder to extract convolutional feature from vehicle image. And we also combine different feature learned by different layer and use PCA[10] to reduce the dimension of the combined feature. Finally softmax layer is attached to the network to perform classification.

II. OUR METHOD

Our network structure is following:

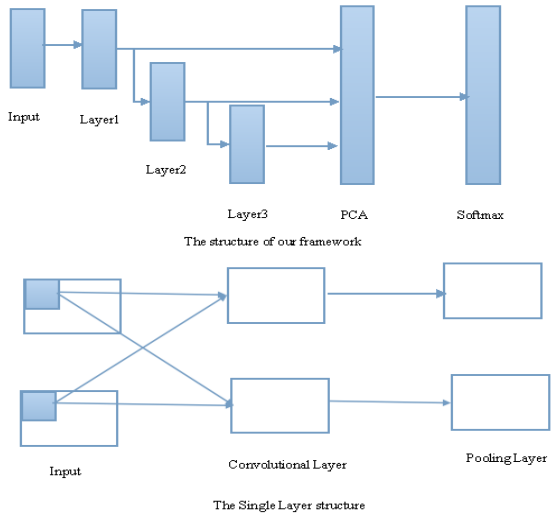

FIGURE I. OUR STRUCTURE.

Our feature learning network consists of five layers including three feature learning layers, a PCA layer and a classification layer. A single feature learning layer includes a convolutional layer and a pooling layer. Different from traditional Convolutional Network, we generate our convolutional kernel by training a Sparse AutoEncoder. Different layer of our network will generate different feature. Then we connect different level feature and use PCA to reduce the feature dimension. Then a softmax layer follows to perform classification.

A. *Sparse AutoEncoder*

Sparse AutoEncoder is an unsupervised feature learning method based on neural network. The basic Sparse AutoEncoder is made up of a input layer with n cells, a hidden layer with k cells and a reconstruction layer with m cells as the

input layer. AutoEncoder can compress the input data to a hidden layer representation, extracting the major information of the input data. Given input data $D=\{x^{(1)},x^{(2)},...,x^{(m)}\}, x^{(i)}\in R^n$, then the encoding procedure and the decoding procedure can be expressed by the followed form:

$$z = f(wx+b) \tag{1}$$

where w_y and w_z is the weight matrix between input layer and the hidden layer and the weight matrix between the hidden layer and the output layer. Accordingly b_y and b_z is the basis vector. The $f(\cdot)$ function is called Activation function by which the nonlinear of network is archived. There are many Activation functions proposed, such as the sigma function, the tanh function and so on. The purpose of AutoEncoder training is to minimize the error between the input layer and the reconstruction layer $\arg\min_{w_y,w_z,b_y,b_z}(c(x,z))$ where given the input, z only depends on the parameters w_y , w_z , b_y , b_z . $c(x,z)$ is the loss function. There are many ways to express the loss. However MSE is a common way to measure the error between the reconstructed data and the input data. So the parameter updating is expressed as

$$\begin{aligned} W &= W-\eta\frac{\alpha\cos t(x,z)}{\alpha W}\\ b_y &= b_y-\eta\frac{\alpha\cos t(x,z)}{\alpha(b_y)}\\ b_z &= b_z-\eta\frac{\alpha\cos t(x,z)}{\alpha b_z} \end{aligned} \tag{2}$$

When training has been performed, we can remove the reconstruction layer and the response value of the hidden layer is the code of the input. The code can be used to perform classification or used as the input of higher layer. At the same time, it is necessary to attach the restriction $\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}(W_{ij})^2$ to avoid overfitting to make the weight matrix value too high. And Sparse AutoEncoder append the KL distance on the loss function to make sure that most of the hidden layer cells values are almost zeros and only a few are activated. So the final Sparse AutoEncoder loss function is as follow:

$$J(W,b)=\left[\frac{1}{m}\sum_{i=1}^{m}(\frac{1}{2}\|h_{W,b}(x^{(i)})-x^{(i)}\|)^2\right]+\frac{\lambda_1}{2}\sum_{l=1}^{L}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}(W_{ji}^{(l)})^2+\beta\sum_{j=1}^{s_2}KL(\rho\|\hat{\rho}_j) \tag{3}$$

Where λ_1 is the weight decay and β is the sparsity constant which is used to control how much the sparsity loss affects the whole loss function. And $\hat{\rho}$ is the average activation of the j hidden cell which can be expressed as $\hat{\rho}_j=\frac{1}{m}\sum_{i=1}^{m}[\alpha_j^{(i)}]$ where $a_j^{(i)}$ is the j hidden cell's response of the i example.

B. Convolutional Feature

When the image size is small, training the Sparse AutoEncoder directly using the whole image is acceptable. But when the image size is vary big, training Sparse AutoEncoder will take a long time. Thanks to the concept of local receive field and weight sharing, Convolutional Network has less parameters to be trained. So We combined the Sparse AutoEncoder and local receive field. In our work, we use Sparse AutoEncoder to get the convolutional kernel. Then Sparse AutoEncoder is used to learn convolutional feature.

The details of our method is as follow:

(1) Randomly sample the same area on all the feature maps to get the train examples and vectorize the examples. The examples are also regularized in this step.

(2) Use the trainning examples to train the single Sparse AutoEncoder to get the Sparse AutoEncoder Model.

(3) Use each row of the Sparse AutoEncoder's weight matrix as the convolutional kernel to learn the convolutional feature from the input.

(4) Perform the pooling operation on the feature maps.

(5) Repeat the above steps several times to get the deep network.

C. Feature Fusing

Although by using the feature learned by the above multi-layer network we can get a relatively good performance, there are some information lost while passing through the multi-layer network. So to get a better performance, we connect different feature from different layer to get a feature with more information. But the connected feature is vary huge. This will make it difficult to operate the combined feature. So we use PCA to reduce the dimension of the feature.

D. Classification

In our work, we adopt softmax as the classifier. In softmax regression model, there are several labels to predict. That means that label y has k different values. When given the train dataset $\{(x^{(1)},y^{(1)}),...,(x^t,y^t)\}$, $y(i)$ can take the value of $1....k$. The softmax function is used to predict the probability that x belongs to the j class. In other word, we want to estimate the probability that x belongs to every class label. So we get a vector with k values to represent k different probability. The predicting function is as follow:

$$h_\theta = \begin{bmatrix} p(y^i = 1 \mid x^i;\theta) \\ . \\ . \\ . \\ \mathrm{p}(y^i = l \mid x^i;\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^{\mathrm{T}} x^i}} \begin{bmatrix} e^{\theta_1^{\mathrm{T}} x^i} \\ . \\ . \\ . \\ e^{\theta_k^{\mathrm{T}} x^i} \end{bmatrix} \quad (4)$$

Where θ is the model parameter, $\sum_{j=1}^{k} e^{\theta_j^T x^i}$ is used to regularized the probability distribution so that the sum of the probability is 1. In conclusion, the cost function of softmax regression is

$$J(\theta) = -\frac{1}{\mathrm{m}} \left[\sum_{i=1}^{m} \sum_{j=1}^{k} 1\{y^{(i)} = j\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}} \right] \quad (5)$$

III. Experiment and Analysis

A. Image Dataset

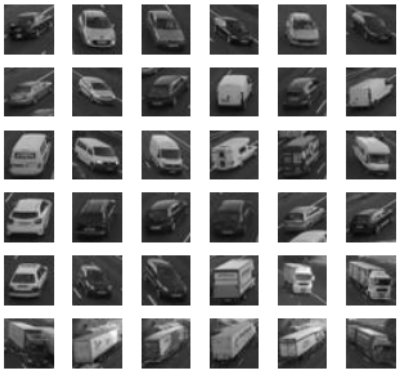

FIGURE II. SOME TRAIN EXAMPLES.

In our work, some used images are from Caltech image database, and the others are collected from the traffic video. Before training , all images are resized to 32*32. And the images are divided into two parts. One is for training, the other is used for testing.

B. Experiment

To validate our method, we compare our method with Stacked AutoEncoder. In our experiment, the Stacked AutoEncoder consists of three layers. Each layer size of our Stacked AutoEncoder is 1024, 300, 300, 300. We also compare the result of method with and without of multi-level feature fusing. Here is the result.

TABLE I. RESULTS OF DIFFERENT FEATURE LEARN METHODS.

	SSAE	Our method without fusing	Our method with fusing
accuracy	0. 7675	0. 8025	0. 8425

Different classification method also affect the final result. To find which classification method is fit to our feature learning method, we compare several classification methods, including softmax regression, support vector machine[11] and deep belief network[12]. The result is as follow:

TABLE II. COMPARE DIFFERENT CLASSIFIERS.

	Softmax	DBN	SVM
accuracy	0. 8425	0. 7241	0. 8145

From Figure1, we can see that our method can get a little higher accuracy than Stacked AutoEncoder Network. And by fusing multi-level feature and using PCA to reduce dimension, the recognition rate can improve further. So we can say that our proposed method can works well.

As we can see in Figure 2, the choice of classification method also affects the result. Softmax classifier get the highest accuracy. Support Vector Machine and DBN does not work so well.

IV. Conclusion

In our work, we combined convolutional operation and pooling operation with Sparse AutoEncoder to build a novel feature learning network and stacked our single layer network to form a deep network. We also fuse multi-level feature and use PCA to reduce feature dimension. According to the experiment result, we can see that our framework works well. But there are still some problems. One problem is that although we use PCA to reduce the dimension, the feature is still too huge to handle. Another problem is that we did not propose a fine-tuning method to adjust the network slightly. This will be our next work.

References

[1] SONG Ke-Chen, YAN Yun-Hui, CHEN Wen_Hui, ZHANG Xu. Research and Perspective on Local Binary Pattern. ACTA AUTOMATICA SINICA, 2013, 39[6]:730—744.

[2] Déniz O, Bueno G, et al. Face recognition using histograms of oriented gradients[J]. Pattern Recognition Letters, 2011, 32(12):1598-1603.

[3] David G Lowe. Distinctive Image Features from Scale-Invariant Interest Points[J]. International Journal of Computer Vision, 2004, 60(2):91-110.

[4] Le N D H, Tran M T. A Robust Unsupervised Feature Learning Framework Using Spatial Boosting Networks[C] Machine Learning and Applications (ICMLA), 2013 12th International Conference on. IEEE, 2013, 2: 507-512.

[5] Jiang X, Zhang Y, Zhang W, et al. A novel sparse auto-encoder for deep unsupervised learning[C] Advanced Computational Intelligence (ICACI), 2013 Sixth International Conference on. IEEE, 2013: 256-261.

[6] Phan H T, Duong A T, Tran S T. Hierarchical sparse autoencoder using linear regression-based features in clustering for handwritten digit recognition[C] Image and Signal Processing and Analysis (ISPA), 2013 8th International Symposium on. IEEE, 2013: 183-188.

[7] Walid R, Lasfar A. Handwritten digit recognition using sparse deep architectures[C] Intelligent Systems: Theories and Applications (SITA-14), 2014 9th International Conference on. IEEE, 2014: 1-6.

[8] Zhu M, Wu Y. A novel deep model for image recognition[C] Software Engineering and Service Science (ICSESS), 2014 5th IEEE International Conference on. IEEE, 2014: 373-376.

[9] Ciresan D C, Meier U, Gambardella L M, et al. Convolutional neural network committees for handwritten character classification[C] Document Analysis and Recognition (ICDAR), 2011 International Conference on. IEEE, 2011: 1135-1139.

[10] Liong V E, Lu J, Wang G. Face recognition using Deep PCA[C] Information, Communications and Signal Processing (ICICS) 2013 9th International Conference on. IEEE, 2013: 1-5.

[11] Nello Cristianini and John Shawe-Taylor. An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press, 2000. ISBN

[12] Chen Y, Zheng D, Zhao T. Adapting deep belief nets to Chinese entity detection[C] Mechatronic Sciences, Electric Engineering and Computer (MEC), Proceedings 2013 International Conference on. IEEE, 2013: 1830-1834.