Advances in Computer Science Research, (ACSR), volume 52
2016 International Conference on Computer Engineering and Information Systems (CEIS-16)

Efficient Re-Clustering for Convolutional Neural Networks

Cheng-Ying Wang, Feng Cen
College of Electronics and Information Engineering
Tongji University
Shanghai, China
E-mail: 1433333@tongji.edu.cn, feng.cen@tongji.edu.cn

***Abstract*-With the rapid development of face recognition, it has been widely applied to various fields and scenes. Due to the unstable light condition, rotation and occlusion, the stability of the recognition performance is still an issue. In this work, we study the VGG model and analyze features' distribution extracted with the VGG model. To achieving good performance, we suggest a t-distribution based VGG model to reduce the dimension of features and re-clustering them. Furthermore, by recording the path of dimension reduction in training phase, the computational complexity of the test phase is decreased. The proposed algorithm is evaluated on public available datasets. The experimental results demonstrate the significant improvement of the proposed approach on both the simplification of feature space and the efficiency of recognition, especially on limited data.**

Keywords-face recognition; t-distribution; convolutional network; re-clustering; dimensional reduction

I. INTRODUCTION

Deep learning is widely used in face recognition, which has attracted the attention of vision researchers. Its self-learning process can extract complex features of high dimensional subspace, which can describe face properly. Most recognition algorithm are based on neural network model, such as GoogleNet[1], VGGNet[2], AlexNet[3], etc. And in the last layer of the model, it achieves classification generally with a linear classifier of Softmax. The classical linear classifier Softmax can still be used in deep learning, because the extracted features in high-dimension have almost linear property. But due to often complex factors in the recognition process, such as illumination, facial expression, rotation and occlusion, compared to the features of iris and fingerprint, facial features are a kind of nonlinear complex structure, even with some noise. In a convolutional neural network, the convolutional layers try to learn filters bank nonlinearly given original images. By using deconvolutional network approach to project a fully trained model down to pixel space, the projections from each layer show the hierarchical nature of the features in the network, responding to entire objects with pose variation, class-specific variation, textures, edges and back to image pixels in the first layer [4]. And the output becomes abstract hierarchically inside the pooling window at the same time [5]. It's known that a dataset consisting of faces is a structure of manifold. Manifold learning algorithms assumes that [6] most of R^n consists of invalid inputs, and that interesting inputs can be approximated well by considering only a small number of degrees of freedom, or dimensions, embedded in a higher-dimensional space. Hence, when the data lies on a low-dimensional manifold, it can be most natural for machine learning algorithms to represent the data in terms of coordinates on the manifold, rather than in terms of coordinates in R^n. Figure 1 [7] shows the manifold structure of a dataset consisting of faces, even the inner dimension of the manifold structure is higher. On one hand, in the process from nonlinear feature extraction to linear classification, with the nonlinear dimensionality reduction method of manifold learning, features in the same subspace cluster together and different classes separate even on limited dataset. The linear classifier also gets identifiable with low-dimensional information. On the other hand, the characteristics in the high-dimensional expression directly affect the mapping in low dimensional space. The completed structure of network can describe subspace properly, which can provide enough information to the method of nonlinear reduction. As a result, the VGG model based on t-distribution can outperform most other methods for linear classification.

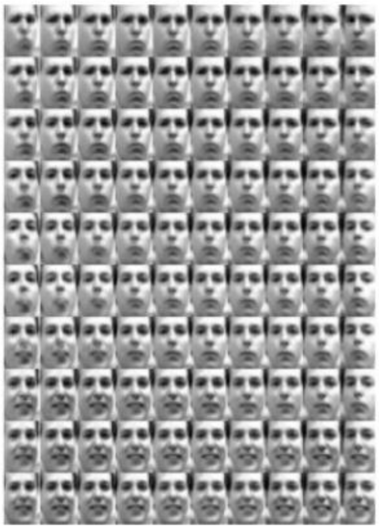

Figure 1. Learned Frey Face manifold: the horizontal row : changes in rotation angle; The vertical column : changes in facial expression.

In recent years, all the proposed deep learning based face recognition methods such as FaceNet[8], Baidu[9], Face++[10] and the series of DeepID[11] have been trained and evaluated on very large wild face recognition datasets, i.e. Labeled Faces in the Wild (LFW). Although the LFW dataset is taken from natural scenes, but compared with other datasets such as AR dataset like Figure 2, the LFW dataset is still limited and be over-simplified. Many works in mid-level face analysis, such as face recognition or facial expression analysis, consider the lower-level face analysis components to be already solved, to avoid low-level difficulties and often designed to focus on mid-level problems (e.g. expression analysis). So the dataset should consider not only realistic scenarios, but also emphasis in collecting diverse and balanced data specially [12]. We use the VGG model to extract features on the LFW and AR dataset, respectively and observe the distance between these features. To ensure the consistency of the size of subspace, each class should have 26 images. As the result shows in Figure 3, the overlap of distances between classes on the LFW dataset is quite small, so individuals are more easily to be separated in high-dimension. While the features distances obtained on AR dataset have more overlapping part. Therefore, the AR dataset are more complex and the inner identifiable information is harder to learn. Overall, the performance of recognition algorithms should be evaluated under a cross and balanced combination of problems including illumination, facial expression, age changes, partial occlusion and some noise corruption. In this way, this kind of designed conditions brings more challenges to face recognition, and compared with the LFW, it is more helpful to evaluate the effect of algorithm.

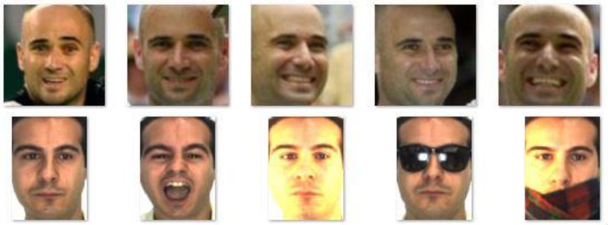

Figure 2. The comparison of subsets in datasets: the first line: LFW dataset; the second line: AR dataset.

In this paper, we propose a t-distribution based VGG model to reduce the dimension of the features and re-cluster the features. Although t-SNE has been used for feature visualization in many applications, the t-distribution, by exactly measuring the similarity between the corresponding points based on distance, reserves the original information and provides to linear classifier Softmax more separable description, whose inner subspace is aggregate simultaneously. Hence, the method is selected to be integrated into the VGG model for realizing dimensional reduction and re-clustering. Moreover, the whole algorithm consists of training and test phases. By recording the path of dimension reduction in training phase, the test process becomes more conveniently. And we also show that although deep learning provides a powerful representation for face recognition, it is hard to achieve the desirable results against expression, age, illumination, and occlusion. To enable deep learning models achieve better results, either of these variations should be taken into account.

The rest of the paper is organized as follows. Section 2 covers a review of existing deep learning methods for face recognition and manifold learning. Section 3 describes the basic principles of two methods and explains the proposed method's detailed characteristic. Section 4 presents the designed experiments and their results. Finally, Section 5 concludes the paper with the summary and discussion of the conducted experiments and implications of the obtained results.

II. METHOD DESCRIPTION

In this section, we describe VGG model for face recognition and t-SNE manifold learning method. Furthermore, we discuss the characteristic of proposed model based on t-distribution.

A. VGG Model

VGG-Face[13] is a deep convolutional network proposed for face recognition using the VGGNet architecture [14]. It is trained on 2.6 million facial images of 2,622 identities collected from the web. The network involves 16 convolutional layers, five max-pooling layers, three fully connected layers, and a final linear layer with Softmax activation. VGG-Face takes color image patches of size 224 × 224 pixels as the input and utilizes dropout regularization [28] in the fully-connected layers. Moreover, it applies ReLU activation to all of its convolutional layers. Spanning 144 million parameters clearly reveals that the VGG network is a computationally expensive architecture. This method has been evaluated on the LFW dataset and achieved an accuracy of 98.95%.

B. Methematical Framework of t-SNE Manifold Learning

t-SNE method [15] is proposed on the basis of SNE method[1], which most importantly replace the Gaussian distribution with t-distribution to automatically fit the relations between high and low dimensional spaces' similarities. The VGG model provides N high-level representations extracted from a color image patch of size 224×224 pixels. The original point $x_{\mathrm{i}} \in R^D$ is a feature in the original space R^D and reduced to the original dimension E by PCA, where $D = 4096$ is the dimensionality of the data space. Compare the differences between the similarity matrix P in the initial space and the similarity matrix Q in the mapping space through an iterative approach, which makes the map point $y_{\mathrm{i}} \in R^F$ belongs to the same subspace aggregate and different classes separate. Hence, the points x_{i}, y_i influence each other and both of them can describe the original input image of face. Let's $|x_i - x_j|$ be the Euclidean distance between two original points, and $|y_i - y_j|$ the distance between the map points. First define a conditional similarity between the two original points as follows:

$$p_{j|i} = \frac{\exp\left(-\left|x_i - x_j\right|^2 \big/ 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\left|x_i - x_k\right|^2 \big/ 2\sigma_i^2\right)} \tag{1}$$

where σ_i^2 is a given variance considering Gaussian distribution around x_i. And to ensure the symmetry of the conditional similarity, the similarity is defined as:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N} \tag{2}$$

And apply the same idea as for the original points to the map points, but with the different distribution t-distribution instead of Gaussian distribution. In this way, the distribution of features in low-dimensional space will be more properly rather than too concentrated.

$$q_{ij} = \frac{\left(1+\left\|y_i - y_j\right\|^2\right)^{-1}}{\sum_{k\neq l}\left(1+\left\|y_k - y_l\right\|^2\right)^{-1}} \tag{3}$$

So the similarity matrix Q between the map points $\left|y_i - y_j\right|$ is defined as:

$$q_{ij} = \frac{f\left(\left|y_i - y_j\right|\right)}{\sum\limits_{k\neq i} f\left(\left|y_i - y_k\right|\right)} \quad , \quad f(z) = \frac{1}{1+z^2} \tag{4}$$

In consideration of the t-distribution's characteristic, for $p_{ij} = q_{ij}$ as Figure 4 shown, the distance between points which are closer in original space becomes smaller, while in the opposite condition it become even farther. It precisely meets our needs for re-clustering in the process of mapping.

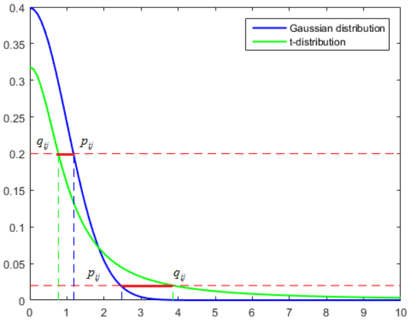

Figure 4. The similarity and distance between corresponding points in high and low dimensional space. In high dimensional space we refer to the Gaussian distribution while in low dimensional space refer to t-distribution.

Define the cost function with the the Kullback-Leiber divergence between the two distributions p_{ij} and q_{ij} as follows:

$$C = \sum_i KL\left(P_i \,\|\, Q_i\right) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}} \tag{5}$$

To minimize this score by the gradient can be computed as:

$$\frac{\delta C}{\delta y_i} = 4\sum_j \left(p_{ij} - q_{ij}\right) g\left(\left|x_i - x_j\right|\right)\left(y_i - y_j\right) \tag{6}$$

Whereas the data similarity matrix p_{ij} is fixed, the map similarity matrix q_{ij} depends on the map points. What we want is for these two matrices to be as close as possible. This would mean that similar data points yield similar map points.

C. The VGG Model Based on T-distribution

In view of the fact that the VGG model has complete self-learning skills, moreover, t-distribution can re-cluster on it properly. As the randomicity of t-distribution, we modify the t-distribution into two modes to apply for VGG model, which can record the path of training mode and keep the dimension of test mode reduce in the same way.

```
Algorithm 1: Training Mode
Input data: $x_i \in R^{N\times D}$, E, F, perplexity.
Output data: $y_i \in R^{N\times F}$, r.
Initialization:
    Step 0. k=0, E=50, F=2, perplexity=30, $y_i \in R^{N\times F}$.
    Step 1. PCA: $x_i \in R^{N\times D} \rightarrow x_i \in R^{N\times E}$
    Step 2. Form the similarity matrix $P$ by $x_i$ in the original space with the Gaussian distribution.
Repeat iterations:
    k:= k+1;
    Step 3. Solve Eq.4 to form the similarity matrix $Q$ in the map space with the t-distribution.
    Step 4. Calculate $grads \in R^{N\times N}$ by $y_i$ with $P$, $Q$.
    Step 5. Get increment matrix as incs by grads, gains; Add it to $y_i : y_i = y_i + incs$; Record gains matrix as $r \in R^{N\times 1}$.
while k<1000.
```

The training mode utilizes the whole training set $x_i \in R^{N\times D}$ to do dimensional reduction after the VGG model extracting features of them by the FC7 layer. The test mode uses the gains matrix recorded in each iteration of the training process and iterate only to the test object as follows:

```
Algorithm 2: Test Mode
Input data: $x_t, x_j \in R^{N\times D}$, E, F, perplexity, r, $y_j \in R^{N\times F}$.
Output data: $y_t \in R^{1\times F}$.
Initialization:
    Step 0. k=0, E=50, F=2, perplexity=30, $y_t, y_j \in R^{N\times F}$.
    Step 1. PCA: $x_t, x_j \in R^{N\times D} \rightarrow x_t, x_j \in R^{N\times E}$
    Step 2. Solve Eq.2 to form the similarity matrix $P$ by $x_t$, $x_j$ in the original space with the Gaussian distribution.
Repeat iterations:
    k:= k+1;
    Step 3. Form the similarity matrix $Q$ by $y_t, y_j$ in the map space with the t-distribution.
    Step 4. Calculate $grads \in R^{N\times 1}$ with $P$, $Q$ only to the $y_t$.
```

Step 5. Get increment value as *incs* by *grads*, *r=gains(iter)* ; Add it to y_t : $y_t = y_t + incs$.

while *k<1000.*

We use the training mode algorithm for recording the path of dimensional reduction, while the test mode for identifying the object by the path. Figure 5 shows the performance of the modified model. We choose an image belongs to the class 005 as a test object, and compare the differences between the two modes. As the results shows, the path is recorded completely and the test object is divided into the right class.

For further exploring, we conduct experiments of the contrast of the two modes on different classes. As it can be observed from Table 1, the error of test object in the modified model, compared with the dispersion of classes, can be ignored. Moreover, the reduction of computation time makes face recognition more efficient.

TABLE I. THE PERFORMANCES OF MODES ON AR

Classes Num	Mode	Coordinate	Time (s)	Error(%)
15	Training	-23.2495,4.2785	2.59	0.027
	Test	-23.2498,4.2787	1.68	
100	Training	-47.3746,-60.5462	237	0.002
	Test	-47.3731,-60.5358	164	

III. EXPERIMENTS AND RESULTS

This section describes the proposed method's detailed characteristic and presents the designed experiments and their results.

A. *Re-clustering Analysis*

The t-distribution based VGG model has effect on both similar and different classes. Since the aim of this experiment is to benchmark the re-clustering of the t-distribution based VGG model against expression, illumination and occlusion, choose the AR as dataset. First of all, Let's see the inner change of features belong to the same class when we reduce the dimension of features from original space 4096D to the space of 512D, 128D, 32D and 2D. In different conditions as Figure 6 displays, we compare the features with the eigenface to see the changes in subspace, where the eigenface is zero point. It can be observed that features approach the zero point, when the dimension decreases by our proposed model. This indicates that the t-distribution do have improvement on the subspace aggregating.

Again the performance is very similar to inner changes. To observe the changes of features belong to different classes, we choose the features of 8 classes and map them from 4096D to 3D by PCA and t-SNE respectively. One of the observation that can clearly be made from Figure 7 is that the features through PCA separate partly, but some of them still mix together. Compared with it, t-SNE can make the distribution of different classes more properly, which provide more identifiable information for linear classifier and the low-dimensional features optimize the efficiency of face recognition.

B. *Face Recognition Experiments*

Effective method of dimensional reduction is not only that different representations of data, but also to obtain effective information of features which can enhance the classification performance. Compared to the PCA algorithm of linear combination of global features, t-SNE algorithm can work on the condition that features are nonlinear and exist in subspaces. The experiments are implied on the AR dataset, which contains face images of size 768×576 pixels with different facial expressions, illuminations, and occlusions from 100 subjects. Each subject of 26 images are separated to training set (21 images) and test set (5 images). The contrast experiments of t-SNE and PCA are based on that the face image is drawn directly into the feature matrix to do dimension reduction and classification, that is, the original feature is 120*165=19800 dimension.

TABLE II. SUMMARY OF THE RESULTS

Method	512D	128D	16D	8D
t-SNE+softmax	1.40%	0.60%	1%	1.40%
PCA+softmax	93.20%	89.80%	31.60%	16.20%
DCNN+t-SNE+softmax	98.80%	99.00%	99.00%	94.60%
DCNN+PCA+softmax	79.40%	76.80%	32.80%	18.20%

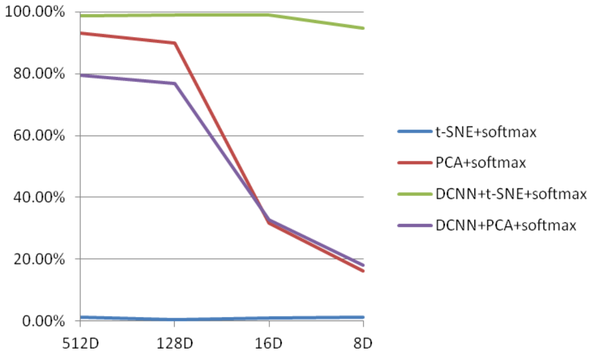

Figure 8. Summary of the results for face recognition.

As the result shows, we can learn that:

- To reduce dimension directly, the t-SNE algorithm can't classify properly.
- But significant improvement is obtained after we adapt the t-SNE algorithm to VGG model. It indicates that VGG model provides enough completed information to the t-SNE algorithm and the t-distribution re-cluster the features who belong to different subspaces exactly.
- Considering the VGG model's accuracy of 98.95% on the LFW, our proposed model maintain excellent recognition characteristics on the more complex dataset and provide more concise features to linear classifier.

Overall, the t-distribution based VGG model reduce the dimension of features in training set and test set, which provide a more identifiable and concise face features representation, then on the base of these, it still improve the recognition accuracy effectively.

IV. CONCLUSION

In this paper, adapting t-distribution into VGG model to achieve dimensional reduction and re-clustering is presented. Because of the nonlinear structure in face feature space, the features extracted with the deep learning network are processed by employing the nonlinear t-SNE manifold learning. It provides linear classifier with identifiable and low-dimensional information. As a future work, although the complexity of network structure is studied, we will investigate the describable features that can be extracted on limited dataset for efficient face recognition.

REFERENCES

[1] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]. IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2014:1-9.

[2] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.

[3] Krizhevsky A, Sutskever I, Hinton G.E. ImageNet Classification with Deep Convolutional Neural Networks[J]. Advances in Neural Information Processing Systems, 2012, 25(2):2012.

[4] Zeiler M D, Fergus R. Visualizing and Understanding Convolutional Networks[M]. Computer Vision – ECCV 2014. Springer International Publishing, 2013:818-833.

[5] Zhou B, Khosla A, Lapedriza A, et al. Object Detectors Emerge in Deep Scene CNNs[J]. Computer Science, 2014.

[6] Goodfellow I, Bengio Y, Courville A, et al. Deep Learning[M]. MIT Press. 2016:160-164.

[7] Kingma D.P and Welling M. Auto-encoding variational bayes[J]. In Proceedings of the International Conference on Learning Representations, 2014.

[8] Florian Schroff, Dmitry Kalenichenko, and James Philbin.FaceNet: A Unified Embedding for Face Recognition and Clustering.Computer Vision and Pattern Recognition (CVPR), 2015.

[9] Liu J, Deng Y, Bai T, et al. Targeting Ultimate Accuracy: Face Recognition via Deep Embedding[J]. 2015.

[10] Zhou E, Cao Z, Yin Q. Naive-Deep Face Recognition: Touching the Limit of LFW Benchmark or Not?[J]. Computer Science, 2015.

[11] Sun Y, Liang D, Wang X, et al. DeepID3: Face Recognition with Very Deep Neural Networks[J]. Computer Science, 2015.

[12] Escalera S, Torres M, Martinez B, et al. ChaLearn Looking at People and Faces of the World:Face Analysis Workshop and Challenge 2016[C]. IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2016:706-713.

[13] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In British Machine Vision Conference, 2015.

[14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. Computing Research Repository (CoRR), 2014. arXiv: 1409.1556.

[15] van der Maaten, L. and Hinton, G. Visualizing data using t-sne[J]. Journal of Machine Learning Research, 2008,9:2579-2605.

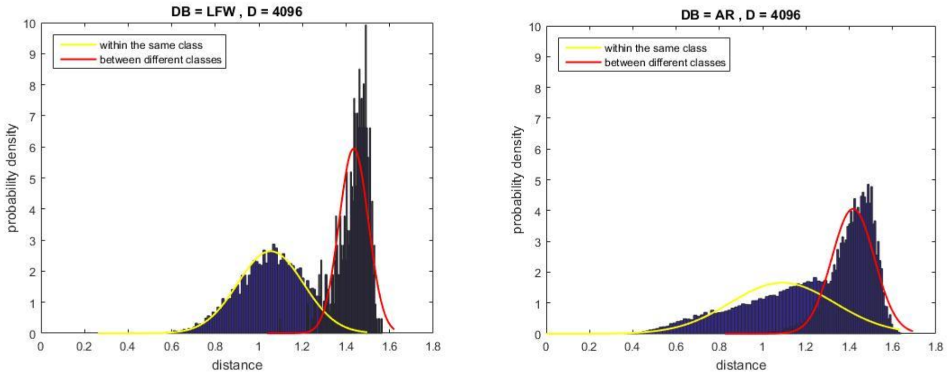

Figure 3. The analysis of distance between different classes in 4096-dimensional space based on the LFW (left) and AR (right) datasets.

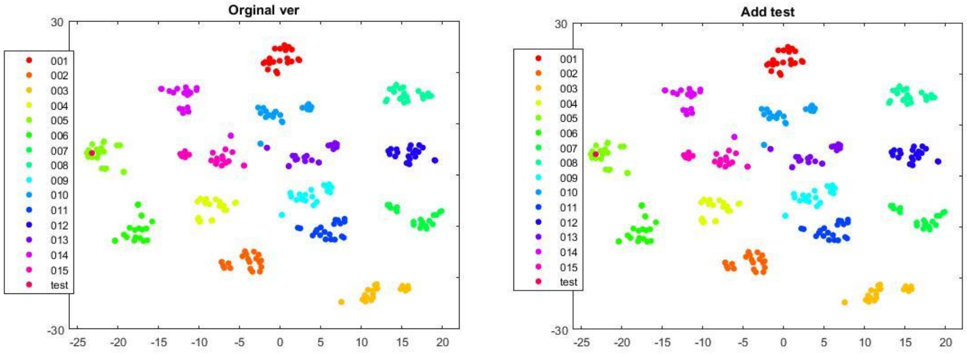

Figure 5. The 2D map space of training model (left) and testing model (right).

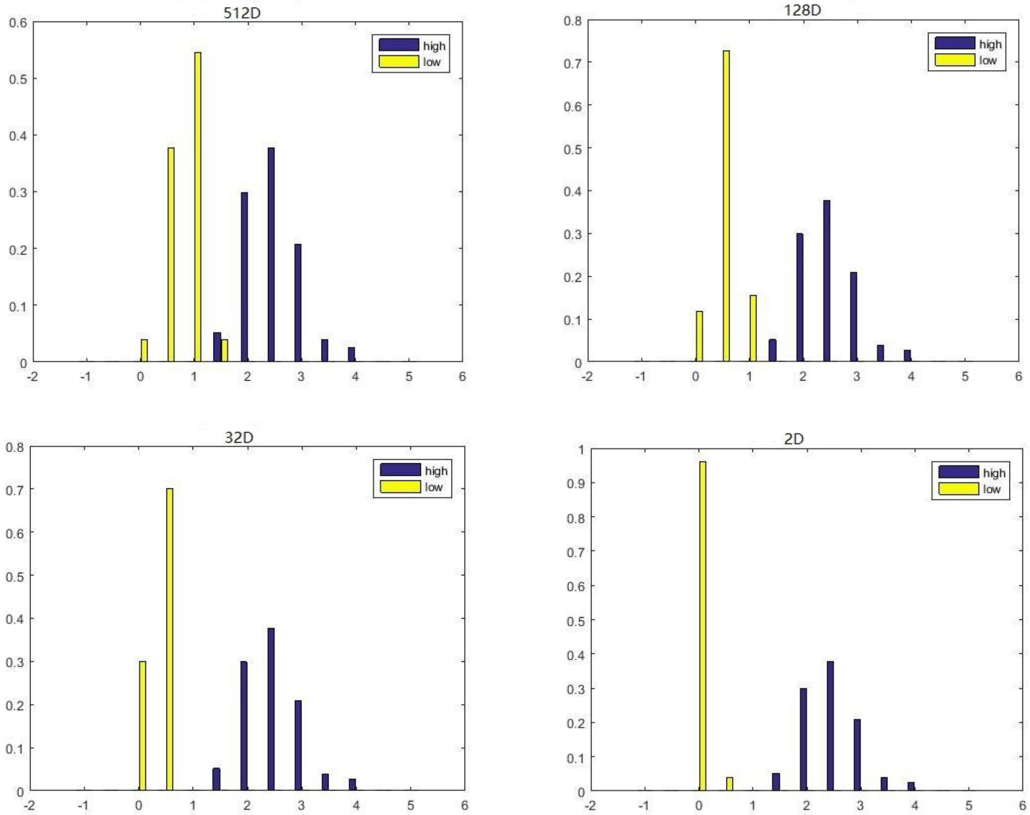

Figure 6. The distribution of features in the same class when dimension decreases: The blue columns represent features in the original space while the yellow ones represent features after dimensional reduction by our proposed model.

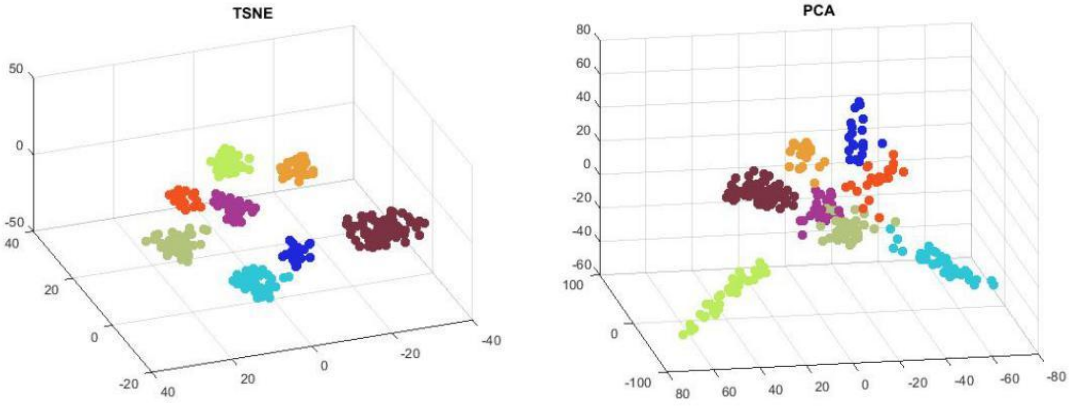

Figure 7. The distribution of different classes in 3D space by t-SNE (left) and PCA (right).