

Unsupervised Spectral Regression Learning for Pyramid HOG

Qiang Li, Zhongli Peng, Xiaomei Lin*

Changchun University of Technology, Changchun, Jilin 130012, China

Abstract

Applying the original raw data to machine learning will bring in a poor performance, because so many features are not necessary and redundant. Extracting a small number of good features will be an important issue, and it can be solved by using dimensionality reduction techniques. However, the popular dimensionality reduction method will suffer from the eigen-decomposition of dense matrix problem which is expensive in memory and time. We adopt unsupervised (unlabeled) spectral regression method for dimensionality reduction, which well avoids the problem of dense matrix eigen-decomposition problem and can be applied on large scale data sets. Histograms of Oriented Gradients (HOG) are robust features which not only well characterize the local shape and appearance but also show a certain degree of local optical and geometry invariance. In order to characterize the local shape and appearance better, we extract a three-tier pyramid HOG descriptor vector for one sample. Then we adopt the unsupervised spectral regression method for dimensionality reduction on these descriptor vectors. Our algorithm can be applied in the library entrance guard system of university and other research fields. Several experiments on well-known face databases have shown good performance and good invariance against illumination, occlusion and local deformation, etc.

Keywords: Dimensionality Reduction; Eigen-decomposition of Dense Matrix; Three-tier Pyramid HOG

1 Introduction

The original raw data in reality is often high-dimensional, which is expensive in memory and time without performing the dimensionality reduction operation. Spectral methods, a powerful tool for dimensionality reduction, aim to acquire the information resided in the eigenvectors of a neighborhood graph matrix of data sets to reveal the low dimensional structure in high dimensional data. However, the problem of eigen-decomposition of dense matrices will be brought in, which is expensive in memory and time. The unsupervised (unlabeled) spectral regression method well avoids this problem and will be well applied on large scale data sets [1].

How to extract robust features is an important issue in the field of machine learning, computer vision, pattern recognition, etc. The Histograms of Oriented Gradients (HOG) performs

*Corresponding author.

Email address: 27453582@qq.com (Xiaomei Lin).

quite well in characterising the local shape and appearance of object by capturing the gradient structure. What's more, HOG will show a certain degree of invariance against the local optical and geometrical deformation [2]. HOG will perform a local normalization operation on the block. The final result will be affected by the size of cells. So, in order to characterize the local shape and appearance better, we construct a three-tier pyramid for each sample, and extract the three-tier pyramid HOG features for each sample. Finally, we will carry out an unsupervised spectral regression learning for the three-tier pyramid HOG.

The rest of this paper is organized as follows: In Section 2, we introduce the unsupervised Spectral Regression Learning. Pyramid Histograms of Oriented Gradients (HOG) is reviewed in Section 3. In Section 4, we propose our Unsupervised Spectral Regression Learning for Pyramid HOG. Then, experiments and results is presented in Section 5, and followed by the conclusion in Section 6.

2 Unsupervised Spectral Regression Learning

The spectral embedding algorithms aim to seek a linear function which minimizes the object function for spectral embedding learning. These algorithms will suffer from the high computational time cost and expensive memory requirements, when it comes to the eigen-decomposition of dense matrices. So, these algorithms won't be suited to the large scale data sets. And, the data in reality is always unlabeled. We adopt the unsupervised spectral regression for dimensionality reduction, which avoids the eigen-decomposition of dense matrices. The rough steps of the unsupervised spectral regression learning are as follows [1].

1. Solve the generalized eigenvalue problem of the following function to get y , the eigenvectors in accordance with the maximum eigenvalues.

$$\hat{y} = \underset{y^T D y = 1}{\operatorname{argmax}} y^T S y = \underset{y^T D y}{\operatorname{argmax}} \frac{y^T S y}{y^T D y} \Rightarrow S y = \lambda D y \quad (1)$$

2. Use the technique to solve least square problem [3], such as LSQR to find the projection matrix a which satisfies $X^T w = y$ [4], and more importantly these efficient iterative algorithms can well handle large scale least square problems.

$$w = \underset{w}{\operatorname{argmax}} \left(\sum_{i=1}^N (w^T x_i - y_i)^2 + \alpha \|w\|^2 \right) \quad (2)$$

$\alpha \|w\|^2$ the a penalty term, which can solve the ill posed minimization problem of the least square when the number of samples is smaller than the number of features. This is the regularized least square called ridge regression [5].

3 Pyramid Histograms of Oriented Gradients (HOG)

Histograms of Oriented Gradients (HOG) can well characterize the local shape and appearance of the objects. What's more, they show a certain degree of local optical and geometry invariance. But, we know that the size of the "cell" impacts an important role in fully playing the performance

of the HOG. In order to characterize the local shape and appearance better, we extract a three-tier pyramid HOG descriptor vector for one sample. The roughly steps are as follows [2].

1. Gamma Normalization. Apply the compression formula to the face images and get the grayscale images.

2. Image gradient calculation. Take the derivative at the position of each pixel. We will get the gradient in the abscissa and ordinate directions and obtain the magnitude of the gradient and the gradient orientation. The derivative operations can well capture the contour and some texture information, but also weaken the effects of illumination. We will obtain the image gradient for each image on the three-tier pyramid (see Fig. 1).

3. One-dimensional gradient direction histogram construction. Collecting the gradient information for each “cell”, such as with the size 8×8 pixel, 9 bins and orientation 180 to construct the one-dimensional gradient direction histogram for each image on the three-tier pyramid.

4. Cells combining into block and local contrast normalization implement for each block. Such as combining the 2×2 cells into one block and choose the L2-Hys for the normalization scheme.

5. Descriptor formation. Gather these one-dimensional gradient direction histograms for each image on the three-tier pyramid into the final descriptor vector.

The rough steps are demonstrated in Fig. 1.

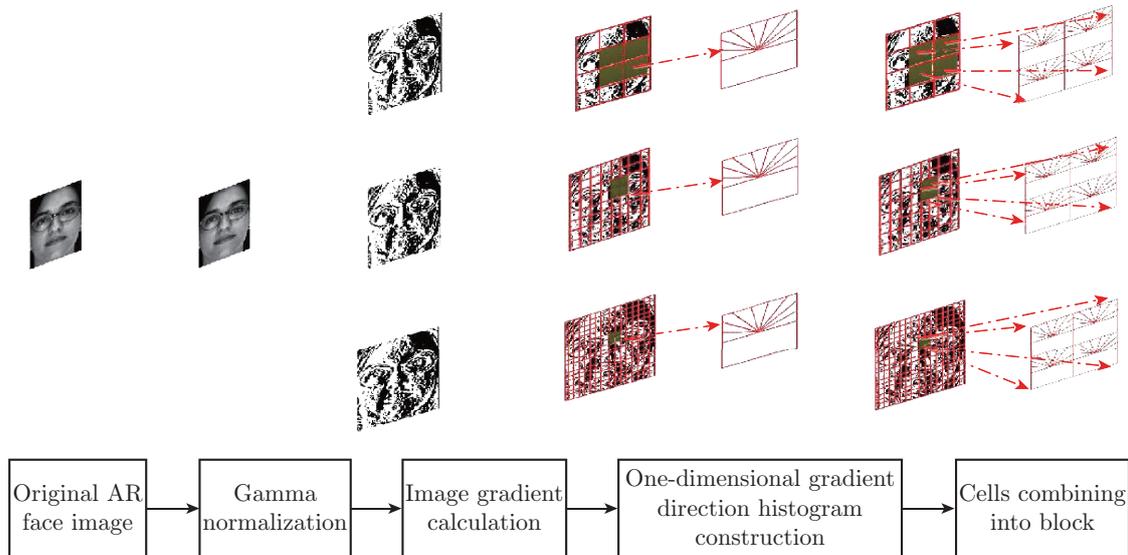


Fig. 1: The rough steps of extracting a three-tier pyramid HOG features; From the top to bottom, the first tier: cell size 40×30 , block 2×2 , bins 9, overlap 0.5; The second tier: cell size 20×15 , block 2×2 , bins 9, overlap 0.5; The third tier: cell size 10×8 , block 2×2 , bins 9, overlap 0.5

4 Unsupervised Spectral Regression Learning for Pyramid HOG

In this section, we first describe the process of extracting the three-tier pyramid HOG features and then we implement the unsupervised spectral regression learning for the three-tier pyramid HOG features.

4.1 Three-tier Pyramid HOG Features

In order to characterize the local shape and appearance better, we construct a three-tier pyramid for each sample, and extract the three-tier pyramid HOG features for each sample. The specific steps are as follows [2].

1. Gamma Normalization is as shown in Section 3.

2. Image gradient calculation. The common method is to make use of the $[-1 \ 0 \ 1]$ gradient operator for convolution operation. Then, we will get the gradient in the abscissa and ordinate directions, obtaining the magnitude of the gradient and the gradient orientation, as is demonstrated in the following equations.

$$G_x(x, y) = I(x + 1, y) - I(x - 1, y) = [-1 \ 0 \ 1] * I(x, y) \quad (3)$$

$$G_y(x, y) = I(x, y + 1) - I(x, y - 1) = [-1 \ 0 \ 1] * I(x, y) \quad (4)$$

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \quad (5)$$

$$\theta(x, y) = \arctan(G_y(x, y)/G_x(x, y)) \quad (6)$$

3. One-dimensional gradient direction histogram construction. This operation is carried out for the local image. We collect the gradient information into the one-dimensional histogram, as is revealed in Fig. 2 and Fig. 3. In this paper, we choose the unsigned gradient.

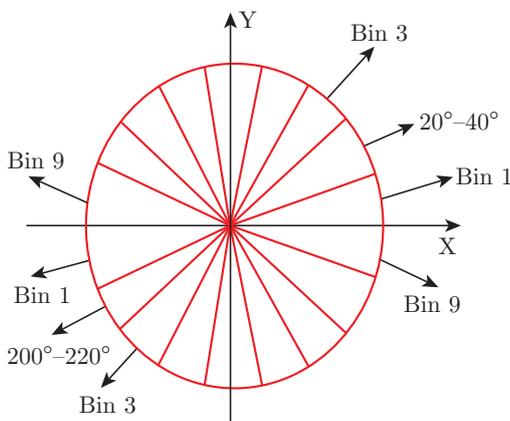


Fig. 2: One-dimensional gradient direction histogram construction: unsigned gradient ($0^\circ - 180^\circ$)

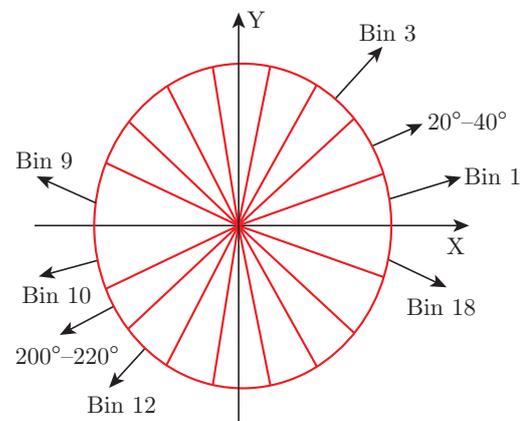


Fig. 3: One-dimensional gradient direction histogram construction: signed gradient ($0^\circ - 360^\circ$)

4. Cells combining into block and local contrast normalization implement for each block. The local contrast normalization is essential to weaken the adverse effect of the local illumination and foreground-background contrast changes which make the gradient intensity range is very large. So, we combine the smaller cells, such as 2×2 cells into the larger block on which we can implement the L2-Hys normalization and these blocks are partially overlapped, such as overlap is 0.5.

5. Descriptor formation. Gather these one-dimensional gradient direction histograms into the final descriptor vector.

4.2 Unsupervised Spectral Regression Learning for the Three-tier Pyramid HOG Features

The raw data sets are often unlabeled, and the spectral regression learning can well avoid the eigen-decomposition of dense matrices. So, we adopt the unsupervised spectral regression learning for the three-tier pyramid HOG features to reduce the dimensionality. The procedure of this algorithm is stated as follows [1].

1. Building the affinity graph. Let G^{hog} denote the graph with N nodes, $X^{hog} = \{x_1^{hog}, x_2^{hog}, \dots, x_i^{hog}, \dots, x_N^{hog}\}$ be the three-tier pyramid HOG descriptor vector sets, among them the i -node corresponds the three-tier pyramid HOG descriptor vector x_i^{hog} . We will connect the node i and node j , if they are among the K nearest neighbors of each other.

2. Setting the weights. We can use the simple-minded and the heat kernel methods to evaluate the weights.

3. Response generation. Getting the d eigenvectors by solving the generalized eigen problems of Eq. (7) corresponding the largest d eigenvalues. $Y^{hog} = \{y_1^{hog}, y_2^{hog}, \dots, y_d^{hog}\}$. It is straightforward to show that the first eigenvector is a vector of all ones with eigenvalue 1 [6].

$$Sy^{hog} = \lambda Dy^{hog} \quad (7)$$

4. Regularized least squares. The minimization problem of the least square will be ill posed when the number of samples is smaller than the number of features. We adopt the following regularized least squares to find the $d - 1$ vector $\{w_1^{hog}, w_2^{hog}, \dots, w_{d-1}^{hog}\}$.

$$w = \underset{w}{argmax} \left(\sum_{i=1}^N (w^T x_i - y_i)^2 + \alpha \|w\|^2 \right) \quad (8)$$

5. USR Embedding. Let $W = \{w_1, w_2, \dots, w_{d-1}\}$ be the projection matrix. We can get the $d - 1$ dimensional subspace by the following embedding.

$$x^{hog} \rightarrow z^{hog} = W^T x^{hog} \quad (9)$$

5 Experiments and Results

5.1 Experiment Results on the AR Face Database

The AR face database contains 126 individuals (both males and females), with each person 26 images taken in two sessions, separated by two weeks (14 days) time, over 4 000 images. Different facial expressions, illumination conditions and occlusions are included in this database [7]. In this experiment, we choose the former 20 males and former 20 females with each person the former 13 images to build our sub AR face database, totally 520 face images. There are 6 groups of training and testing sets, $(G2/P11, \dots, G7/P6)$. Among them G_m/P_n denotes m images for training and n images for testing. We iterate 20 times for each training and testing set and take the average value as the error rate. The second person of our sub AR face database is exhibited in Fig. 4. The error rate of the 6 groups of training and testing sets is demonstrated in Table 1 and Fig. 5.



Fig. 4: The second person of our sub AR face database

Table 1: Error rate on our sub AR face database; Error rate (%) [dim] \pm std (%)

Algorithm	G3/P10	G4/P9	G5/P8	G6/P7	G7/P6	G8/P5
Original USR	59.60 [80] \pm 3.79	54.90 [107] \pm 4.14	51.14 [134] \pm 3.35	50.00 [155] \pm 4.27	45.04 [179] \pm 2.89	44.85 [206] \pm 3.93
PHOG-USR	57.55 [74] \pm 1.95	50.53 [107] \pm 3.10	43.94 [134] \pm 3.54	38.21 [158] \pm 3.74	30.77 [173] \pm 3.57	26.70 [188] \pm 4.10

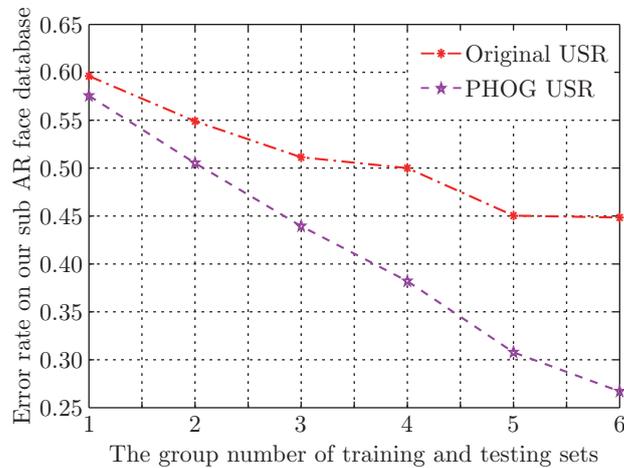


Fig. 5: Error rate on our sub AR face database

5.2 Experiment Results on Yale Face Database and the Robust Test Against Local Distortion

The Yale face database contains 165 grayscale images of 15 individuals with 11 images per person. The expression changes, illumination and posture variation and wearing glasses will appear in this database [8]. We have 6 groups of experiment data ($G2/P9, \dots, G7/P4$), and take the average of 50 times iteration as the final error rate. As is demonstrated in Table 2 and Fig. 7.

Table 2: Error rate on Yale face database; Error rate (%) \pm std (%) [dim]

Algorithm	G2/P9	G3/P8	G4/P7	G5/P6	G6/P5	G7/P4
Original USR	38.44 [20] \pm 5.20	33.58 [23] \pm 3.52	29.28 [35] \pm 3.33	26.42 [47] \pm 3.22	24.64 [56] \pm 3.52	23.17 [50] \pm 3.78
PHOG-USR	30.73 [20] \pm 6.59	24.45 [29] \pm 5.32	17.47 [17] \pm 4.19	10.71 [38] \pm 3.29	5.31 [38] \pm 2.56	3.33 [38] \pm 1.96

In the following, we state the robust test against local distortion of our algorithm. We first perform the local distortion operation for the 1st, 5th and 9th images of each person, as shown

in Fig. 6. We get the error rate as is described above. Table 3 and Fig. 8 will show the results of the 6 groups of experiment data.



Fig. 6: The second person with the 1st, 5th and 9th images distorted of Yale face database

Table 3: Error rate with distortion on Yale face database; Error rate (%) \pm std (%) [dim]

Algorithm	G2/P9	G3/P8	G4/P7	G5/P6	G6/P5	G7/P4
Original USR	40.83 [20] \pm 5.85	35.63 [29] \pm 4.26	30.51 [26] \pm 3.83	28.00 [32] \pm 3.71	25.68 [41] \pm 3.19	24.87 [35] \pm 4.11
PHOG-USR	34.18 [20] \pm 7.25	28.53 [29] \pm 5.92	22.34 [29] \pm 4.99	16.13 [47] \pm 3.18	11.97 [35] \pm 2.29	8.20 [50] \pm 3.41

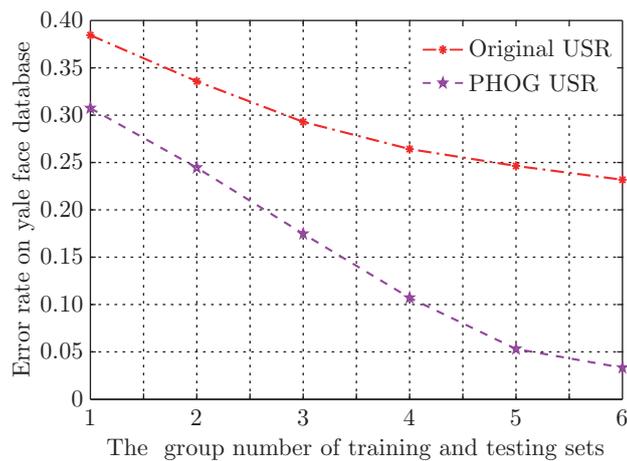


Fig. 7: Experiment results: Error rate on Yale face database

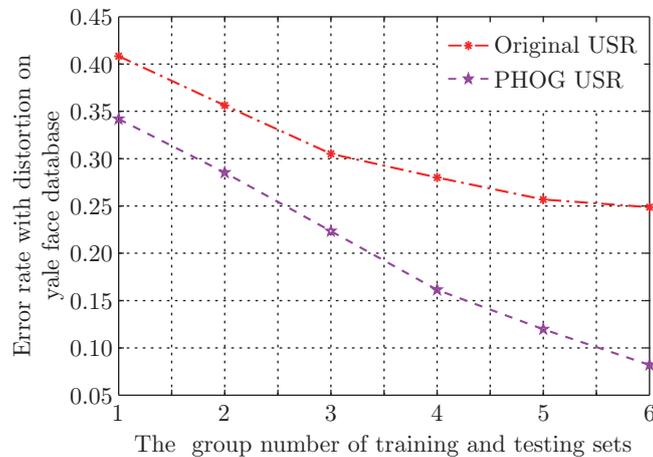


Fig. 8: Experiment results: Error rate with distortion on Yale face database

6 Conclusion

In this paper, we extract the robust feature sets, the three-tier pyramid HOG descriptor vectors which can well characterize the local shape and appearance of face images and possess a certain degree of good invariance against the local optical and geometrical deformation. Then, we adopt the unsupervised spectral regression learning for the three-tier pyramid HOG, which well avoids the problem of eigen-decomposition of dense matrices. The experiment results have shown the good performance of our algorithm and this algorithm can be, for example applied in the library entrance guard system of university. In the near future, I will extend my research into novel fields, such as, gene identification and facial expression recognition.

References

- [1] D. Cai, X. F. He, J. W. Han, Spectral regression for dimensionality reduction, 2007
- [2] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, 1, 2005, 886-893
- [3] G. H. Golub, C. F. V. Loan, Matrix Computations, 3rd edition, Johns Hopkins University Press, 1996
- [4] T. Hastie, A. Buja, R. Tibshirani, Penalized discriminant analysis, *Annals of Statistics*, 23, 1995, 73-102
- [5] T. Hastie, R. Tibshirani, J. Friedman, The elements of statistical learning, Data Mining, Inference, and Prediction, New York: Springer-Verlag, 2001
- [6] F. R. K. Chung, Spectral graph theory, Regional Conference Series in Mathematics, AMS, 92, 1997
- [7] AR Face Database, <http://www2.ece.ohio-state.edu/aleix/ARdatabase.html>
- [8] Yale Face Database, <http://cvc.yale.edu/projects/yalefaces/yalefaces.html>