Proceedings of the International Conference on Machine Vision and Machine Learning
Prague, Czech Republic, August 14-15, 2014
Paper No. 132

Face Recognition with Permutation Coding Neural Classifier Improved with Skewing Transformations

Alejandro J. Ibarra Gallardo, Z. Cruz Monterrosas
Computer Science and Engineering, UNAM
Av. Universidad Nº 3000, Ciudad Universitaria, CP 04510, México City, México
ibarra.aj@comunidad.unam.mx, zamiracm@gmail.com

Ernst Kussul, Tatiana Baidyk
Centro de Ciencias Aplicadas y Desarrollo Tecnológico, Universidad Nacional Autónoma de México,
Circuito Exterior S/N, Ciudad Universitaria, AP 70-186, CP 04510, México D.F, México
ekussul@unam.mx; tbaidyk@unam.mx

Abstract **-** Face recognition is one of the most important tasks in image and pattern recognition. We propose to use Permutation Coding Neural Classifier (PCNC) in the face recognition task. This neural classifier has demonstrated good results in face recognition on the FRAV3D image database, but several experiments were less successful in comparison with Support Vector Machines (SVM). For this reason we decided to generate new distortions as skewing and add them to the training set to increase the initial data base. We used skewing (also called shear transformation) as an additional type of distortions and as a result, we obtained a significant improvement in recognition rate compared with previous results of PCNC and SVM.

Keywords**:** Face recognition, Neural networks, Computer vision, Face recognition, Permutation Coding Neural Classifier (PCNC).

1. Introduction

This article is devoted to the face recognition problem. To test our algorithms we needed a face image database. There are many proposals on the Internet, for example, the ORL image database (Web-1), the FRAV3D image database (Web-2), or the face image database LFW (Web-3). After an analysis of several different face image databases, we selected the FRAV2D. This image database contains images in 2D, 2.5D and 3D. In this paper, we used only images in 2D. One example image of FRAV2D is presented in Fig.1.

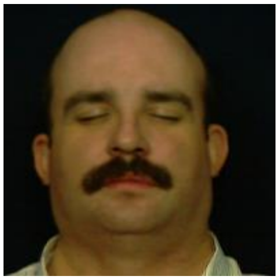

Fig. 1. Image from FRAV2D image database.

The FRAV2D image database contains 105 individuals, mainly young adults, with approximately one woman for every three men (Web-2), (Conde, 2006). There are 16 captures per person in different face poses and/or lighting conditions.

Different methods have been developed to recognize faces, for example, face recognition by elastic bunch graph matching (Wiskott et al., 1997), two-dimensional principal component analysis (Yang et al., 2004), face description with local binary patterns (Ahonen et al., 2006) and others (Zhao et al., 2003). Our proposal is based on the PCNC neural classifier (Kussul et al., 2006, 2010, 2013) which we describe below.

The structure of this article includes the description of the skewing distortions (section 2), the neural classifiers (section 3), the PCNC structure and algorithms (section 4), the description of the experiments and results of face recognition with the skewing distortions (section 5), and finally the conclusion (section 6).

2. Skewing Procedure for Image Distortions

The procedure of image skewing is sometimes known as shear transformation. A good definition of this transformation is provided in Wikipedia (Web-4). In plane geometry, a shear mapping is a linear map that displaces each point in a fixed direction, by an amount proportional to its signed distance from a line that is parallel to that direction. This type of mapping is also called transvection, or just shearing.

Shear transformation can be applied as a horizontal shear, as a vertical shear or as both. An algorithm to perform this transformation is described in detail by Dewald Esterhuizen (Web-5). All of the concepts explored have been implemented by means of raw pixel data processing. The following pair of equations expresses both transformations (over *y* and *x* axes).

$$skewing(x) = x + \sigma y - \frac{W\sigma}{2}, \tag{1}$$

$$skewing(y) = y + \sigma x - \frac{H\sigma}{2}, \tag{2}$$

where *skewing*(*x*) is the result of a horizontal skewing transformation, *x* is the coordinate of the original image, *skewing*(*y*) is the result of a vertical skewing transformation, *y* is the coordinate of the original image, σ is the factor of shearing or skewing and has a value between [0, 1], and finally, *H* and *W* represent height and width of the image in pixels, respectively.

To apply the skewing transformation, a new position of each point of the image is calculated, then at the end of the skewing transformation they are rendered in a single array of points.

The implemented algorithm allows for both horizontal as vertical image transformation by various factors of skewing. In Fig. 2 we present a few examples of skewing transformation using different factors (described in degrees) to the left and right. These images were used to extend the training set of images for the PCNC neural classifier.

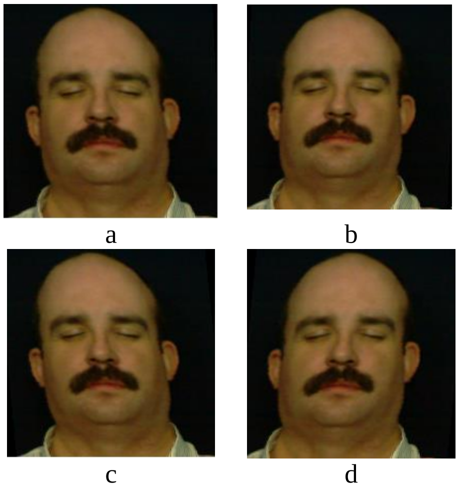

Fig. 2. Skewing factors: a) 5º to the left, b) 5º to the right, c) 10º to the left, d) 10º to the right

3. Neural Classifier

The PCNC (Permutative Coding Neural Classifier) structure (Kussul et al., 2007) is presented in Fig.3.

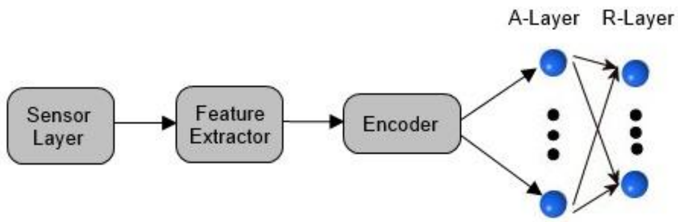

Fig. 3. PCNC structure.

The classifier is based on the concept of Random Local Descriptors (RLD). RLD works as a general feature extractor by making connections between neurons in the associative layer and random points in the input image, and calculating a brightness function of the selected point (Kussul et al., 2006).

The scheme of the neural network recognition system is shown in Fig. 4.

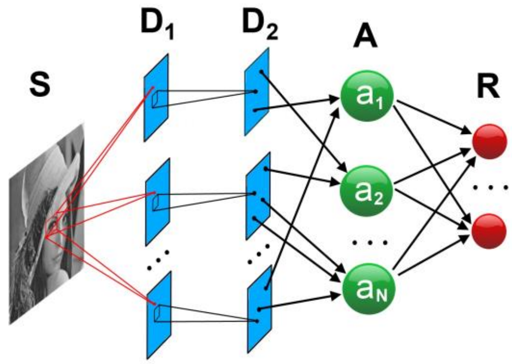

Fig. 4. Structure of general purpose image recognition system.

The system is a multilayer neural network with the Rosenblatt's perceptron concept (Rosenblatt, 1962). *S*-layer (sensor system) is the input image; the D_1-layer contains RLD neurons of the lowest level; and the D_2-layer contains RLD neurons of higher level. *R*-layer contains the output neurons; each of these neurons corresponds to the image class under recognition.

The detailed scheme of RLD is shown in Fig. 5. Neurons 2-5 test de *S*-layer pixels randomly selected from the delimited *H*x*W* rectangle area.

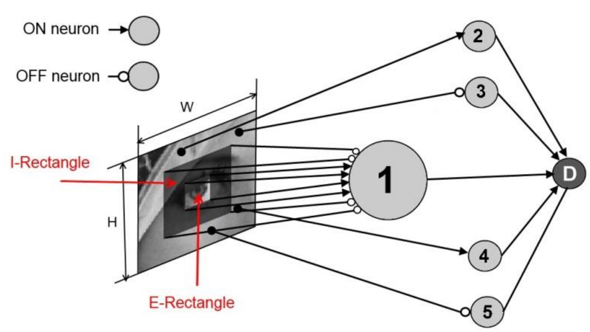

Fig. 5. Detailed RLD Scheme.

We work with different kind of neurons ON, OFF, and a complex neuron. ON neurons select the positive points of interest whereas OFF neurons select the negative ones. RLD neuron outputs are binary, i.e. “1” or active, and “0” or inactive (Fig. 6).

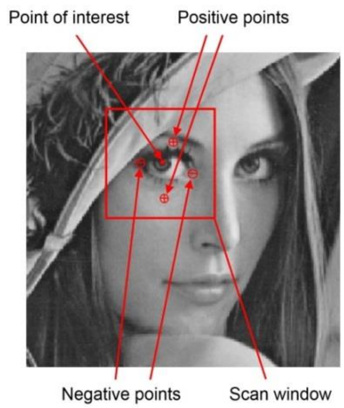

Fig. 6. Points of interest selected by feature extractor.

ON neuron activates if the brightness gradient b_i of the corresponding pixel is higher than the neuron threshold T: $b_i \geq T_i$. OFF neuron activates if the brightness gradient b_i of the corresponding pixel is less than the neuron threshold T: $b_i < T_i$. The threshold value is selected randomly among the input image brightness values $T_{min} \leq T_i \leq T_{max}$.

Neuron 1 is a complex neuron and has excitatory connections with all the pixels inside E-rectangle, and inhibitory connections with the pixels outside the E-rectangle, but inside the I-rectangle. For face recognition task excitatory connections must be inversely proportional to the area of the E-rectangle and inhibitory connections must be inversely proportional to the area of the I-rectangle. Thus, complex neurons detect the most informative points of the image. *D*-neuron or AND-neuron (Fig.5), has an output “1” if and only if all 1-5 neurons are activated.

The topology of the connections between sensor layer and 2-5 neurons is preserved for all *D*-layer planes. Each plane detects the presence of a specific feature in any place of the image. The number of planes corresponds to the number of extracted features.

D_2-layer contains M planes of neurons, each one of them is connected to neurons in D_1-layer. The output of each D_2-layer neuron is “1” if at least one of its connections with D_1-neurons is “1”. D_2 neurons are also called OR-neurons. We say that a feature exists if all positive and negative points are active, otherwise it is absent.

All the neurons of the *A* associative layer (Fig.4) have trainable connections with *R*-layer neurons.

3.1. Feature Encoder

The extracted features of the input image are presented as binary vectors:

$$V = \{v_i\}(i = 1, \ldots, N), \tag{3}$$

where v_i is equal to 0 or 1. For each extracted feature, F_m, the encoder creates an auxiliary binary vector:

$$U = \{u_i\}(i = 1, \ldots, N), \tag{4}$$

where u_i is equal to 0 or 1. U vector contains K 1’s whose positions are randomly selected for each feature F_m, and $K << N$. The generated position list is saved in memory since it is used as a mask of the F_m feature in the permutation process.

U binary vector is permutated in order to take into account the distance between the feature locations. If the distance is small, we say that vectors are strongly correlated, else it is understood that vectors are weakly correlated. The number of permutations depends on the feature location on the image. As a result from the permutation process, a new vector U^* is created. The permutations are made in horizontal and vertical directions. The example of permutation scheme is presented in Fig.7. The process goes as follows: each element of row zero is connected to a free and randomly selected element from the row one. The process repeats until all elements from the both rows are connected with one-to-one connections.

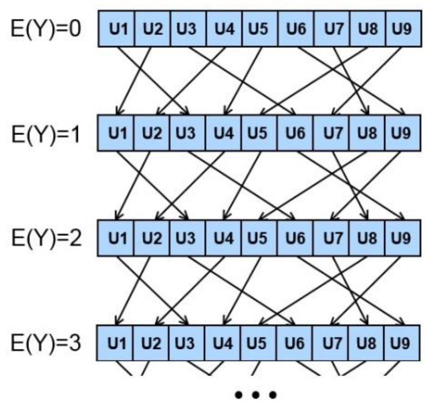

Fig. 7. Permutation pattern for the *Y* coordinate.

In order to code the F_m feature location, a Dc distance is predefined and the following values must be calculated:

$$X = \frac{j}{D_c};\ E(X) = (int)X;\ R(X) = j - E(X) \cdot D_c, \tag{5}$$

$$Y = \frac{i}{D_c};\ E(Y) = (int)Y;\ R(Y) = i - E(Y) \cdot D_c, \tag{6}$$

$$P_x = (int)(R(X) \cdot \frac{N}{D_c}), \tag{7}$$

$$P_y = (int)(R(Y) \cdot \frac{N}{D_c}), \tag{8}$$

where $E(X)$ and $E(Y)$ are the integer part of X and Y, respectively; $R(X)$ and $R(Y)$ are the fractional part of X and Y also respectively; i is the vertical coordinate of the detected feature; j is the horizontal coordinate of the detected feature and N is the number of neurons. $E(X)$ and $E(Y)$ show the number of permutations to perform on the X and Y directions, whereas that P_x and P_y are the number of neurons in the range $[0, N)$ for which an additional permutation is needed (Kussul et al., 2006). The permutations are only performed for non-zero components of U vector.

4. PCNC Structure and Algorithms

In this section we want to explain the PCNC structure and algorithms we worked with. We have termed the small window (Fig.6) and the neuron randomly connected with it as Random Local Descriptor (RLD). The PCNC classifier contains some hundreds of RLDs. But each RLD scans the whole image and if it detects the feature that correspond to this RLD (neuron output is equal to 1), a special coding procedure introduces the V binary vector to A-layer. The binary code of the U^F feature is modified in correspondence to the detected position (x, y) of the feature F. Let us denote this vector as U_{xy}^F. Each vector U_{xy}^F has N dimension equal to the dimension of the V binary vector but it contains only m

components equal to 1 (m<<N). In the experiments presented in this article we used m=16. The binary vector V is composed as follows:

$$V = \bigcup U_{xy}^{F}, \tag{9}$$

where $\bigcup$ is a binary disjunction for all RLDs detected in the image. The number of detected RLDs can significantly vary and depends on concrete images. For this reason we applied a special binding procedure to the V vector for decreasing the number of nonzero components m in the V binary vector.

5. Experiments and Results of Face Recognition

In Table 1 we present the test definition of previous experiments (Kussul et al., 2013). Table 1 show what images we use for training set and what images for test set from FRAV2D face image database. We described in Introduction that FRAV2D contains 16 captures per person in different face poses and/or lighting conditions. So every number in second and third columns of Table 1 corresponds to proper face pose or light condition.

Table 1. Detailed definition of the tests performed with the FRAV2D face image database.

Test Number	Training Set	Test Set
1	3 frontal 1,2,3	1 frontal 4
2	4 frontal 1,2,3,4	1 gesture (smile) 11
3	4 frontal 1,2,3,4	1 gesture (open mouth) 12
4	4 frontal 1,2,3,4	2 illuminations 15,16
5	4 frontal 1,2,3,4	2 rotations 5° φ 7,8
6	4 frontal 1,2,3,4	1 small rotation Z 10
7	4 frontal 1,2,3,4	2 rotations X 13,14
8	4 frontal 1,2,3,4	2 rotations 25° Y 5,6
9	4 frontal 1,2,3,4	1 large rotation Z 9
10	2 frontal, 2 illuminations 1,2,15,16	2 frontal 3,4
11	3 frontal, 1 rotation 5° φ, 1,2,3,7	1 frontal, 1 rotation 5° φ, 4,8
12	3 frontal, 1 rotation 5° φ, 1 illumination 1,2,3,7,15	1 frontal, 1 rotation 5° φ, 1 illumination 4,8,16
13	1,2,3,7,15	4,8,11,16
14	1,2,3,4	11,12
15	1,2	3,4

In Table 2 we show the comparison of results for the PCNC classifier and SVM. In this case we used 12 vertical and horizontal distortions, for example, shifts for several pixels to the left, to the right, up and down of the initial image. The worst results were obtained for experiments with number 6 and 9 (Table 2). So we decided to improve these results with additional distortions in training set.

In Table 3 we present the new results with additional skewing distortions. In this table the second column represents the number of additional images that were skewed and included in the training set. For instance, in the second column, the six first rows mean that the four frontal images were skewed -15º, two frontal images were skewed -10º, two frontal images were skewed -5º, two frontal images were skewed +5º, two frontal images were skewed +10, and four frontal images were skewed +15º. This was done for each person and these skewed images were included in the training set.

Table 2. Comparison of SVM and PCNC results.

Test Number	Errors % PCNC (12 distortions)	Errors % SVM
1	0.78	1.94
2	1.94	5.13
3	1.94	10.68
4	2.45	8.70
5	3.5	14.60
6	33.7	33.93
7	16.4	12.62
8	19.0	27.02
9	54.3	41.14
10	0.59	1.94
11	0.39	4.85
12	0.33	4.17
13	0.44	4.09
14	1.4	8.90
15	0.98	1.46

Table 3. Results of the experiments 6 and 9 considering skewing transformations.

Distortions number	Skewing Image Number (angle)	Errors % (Experiment 6)	Errors % (Experiment 9)
1	4(-15) 2(-10) 2(-5) 2(+5) 2(+10) 4(+15)	25.5	42.2
2	4(-15) 4(-10) 4(+10) 4(+15)	23.5	41.2
3	4(-15) 4(-10) 4(-5) 4(+5) 4(+10) 4(+15)	23.5	43.1
4	4 (-15) 4(-10) 4(+10) 4(+15)	22.5	45.1

The mean value of the error percentage (Table 3) for experiment 6 is 23.75% (without skewing distortions it was 33.7 % for PCNC and 33.93% for the SVM (Table 2)). So the recognition rate is improved. For the experiment 9 the mean value of the error percentage is 42.9%. Without skewing distortions, the error percentage was 54.3% and for SVM it was 41.14% (Table 2). So the recognition rate is improved and almost reaches the SVM results.

6. Conclusion

The results of face recognition from the FRAV3D image database were presented in Kussul et al. (2013). In general, these results were satisfactory, but two experiments with numbers 6 and 9 (Table 2) demonstrated the worse recognition rate. These experiments are connected with lateral inclinations of the head in the test set. In this article we applied new distortions (skewing) to the first images (frontal pose) from the original training set. This procedure allowed us to improve the recognition rate of PCNC and to decrease the error from 54.3% to 42.9% in the experiment with a large inclination angle, and from 33.7% to 23.75% for the one with a small inclination angle. These results are better for small inclination angle and are comparable for large inclination angle with those obtained with SVM.

Acknowledgements

The authors would like to thank the financial support of projects PAPIIT IN102014, IT102814 as well as the Consejo Nacional de Ciencia y Tecnología (CONACYT). We also thank the scientists of University Rey Juan Carlos, in Madrid, Spain for providing FRAV2D face image database, and Adrian Revuelta Cuauhtli for his support in the preparation of this article.

References

Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 28(12), 2037-2041.

Conde, C., (2006). Verificación facial multimodal: 2D y 3D (Doctoral dissertation, Universidad Rey Juan Carlos).

Kussul, E., Baidyk, T., Wunsch, D., Makeyev, O., & Martín, A. (2006). Image recognition systems based on random local descriptors. In Neural Networks, 2006. IJCNN'06. International Joint Conference on (pp. 2415-2420). IEEE.

Kussul, E. M., Baidyk, T. N., Wunsch, D. C., Makeyev, O., & Martín, A. (2006). Permutation coding technique for image recognition systems. Neural Networks, IEEE Transactions on, 17(6), 1566-1579.

Kussul, E., Baidyk, T., & Makeyev, O. (2007, August). Pairwise permutation coding neural Classifier. In Neural Networks, 2007. IJCNN 2007. International Joint Conference on (pp. 1847-1852). IEEE.

Kussul, E., Baidyk, T., & Wunsch, D. C. (2010). “Neural networks and micromechanics” Springer.

Kussul, E., Baidyk, T., Conde, C., de Diego, I. M., & Cabello, E. (2013, August). Face recognition improvement with distortions of images in training set. In Neural Networks (IJCNN), The 2013 International Joint Conference on (pp. 1-6). IEEE.

Rosenblatt, F. (1962). “Principles of neurodynamics” Springer.

Wiskott, L., Fellous, J. M., Kuiger, N., & Von Der Malsburg, C. (1997). Face recognition by elastic bunch graph matching. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 19(7), 775-779.

Yang, J., Zhang, D., Frangi, A. F., & Yang, J. Y. (2004). Two-dimensional PCA: a new approach to appearance-based face representation and recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 26(1), 131-137.

Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face recognition: A literature survey. Acm Computing Surveys (CSUR), 35(4), 399-458.

Web sites:

Web-1: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html, consulted 30 Apr. 2014.

Web-2: http://www.frav.es/, consulted 30 Apr. 2014.
Web-3: http://vis-www.cs.umass.edu/lfw/, consulted 30 Apr. 2014.
Web-4: http://en.wikipedia.org/wiki/Shear_mapping, consulted 30 Apr. 2014.
Web-5: http://code.msdn.microsoft.com/wpapps/Image-Transform-Shear-1a7eca3d, consulted 30 Apr. 2014.