

Face Recognition Performance in Facing Pose Variation

Alexander A. S. Gunawan¹ and Reza A. Prasetyo²

^{1,2}School of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia

Email: ¹aagung@binus.edu

Abstract—There are many real world applications of face recognition which require good performance in uncontrolled environments such as social networking, and environment surveillance. However, many researches of face recognition are done in controlled situations. Compared to the controlled environments, face recognition in uncontrolled environments comprise more variation, for example in the pose, light intensity, and expression. Therefore, face recognition in uncontrolled conditions is more challenging than in controlled settings. In this research, we would like to discuss handling pose variations in face recognition. We address the representation issue using multi-pose of face detection based on yaw angle movement of the head as extensions of the existing frontal face recognition by using Principal Component Analysis (PCA). Then, the matching issue is solved by using Euclidean distance. This combination is known as Eigenfaces method. The experiment is done with different yaw angles and different threshold values to get the optimal results. The experimental results show that: (i) the more pose variation of face images used as training data is, the better recognition results are, but it also increases the processing time, and (ii) the lower threshold value is, the harder it recognizes a face image, but it also increases the accuracy.

Index Terms—Face recognition, pose variation, yaw angles, eigenfaces, principal component analysis

I. INTRODUCTION

NOWADAYS, there are many human jobs that can be replaced by the computer, for example in recognizing the face. The face recognition has many applications in various fields including government and business. Generally these applications can be divided into the two groups: (1) Authentication: to verify users in accessing something, for example, in securing a nuclear reactor [1]. In this application, face recognition has many benefits compared to the traditional passwords, such as preventing forgotten passwords or misuse of stolen keys. The accuracy in this kind of application must be high, so the environments should be controllable. This can be done by restraining the number of people and setting the input images as

the frontal face under steady illumination. (2) Surveillance: to monitor people in the certain area, such as identifying visitors in public services. This application typically has to deal with uncontrolled environments in which input images have background clutters, varying illumination, and large variations in pose. Face recognition, which was developed by Viisage and was deployed at the airport [2], has many false alarms and has finally been terminated. The inaccuracy of this system is mainly due to the uncontrolled environments.

Many researches of face recognition algorithm in the past were developed on controlled face databases in which images have simple backgrounds and little variations in pose, illumination, and expression. Recently researchers have shifted to develop face recognition algorithm in uncontrolled environments. There are three main challenges for face recognition in uncontrolled environments with the variation in expression, illumination, and pose. The first challenge is varying expression, which can reduce recognition performance significantly. Our previous research [3] tried to recognize expression by using Active Appearance Model and Fuzzy Logic. The second problem is illumination variation. It is difficult to recognize the face under varying illumination. Our previous publication [4] proposed Multi-Scale Retinex method to solve illumination problem. Finally, pose variation makes the feature match between two face images from different viewpoints is very difficult. In this research, we would like to discuss handling pose variations in face recognition. We address the pose variation problem by developing a multi-pose of face detection based on yaw angle movement of the head. The recognition and matching steps use the most common methods, which are Principle component analysis (PCA) and Euclidean distance. We argue that these steps are not too crucial in analyzing the pose variation problem that is more related to face detection phase.

The recent researches on pose variation problem [5] reveals that the best frontal face recognition algorithm on Labeled Face in the Wild (LFW) dataset [6] is

poorly in recognizing faces with many pose variations. This algorithm is called as High Dimensional LBP [7] which uses a high dimensional feature based on a single-type Local Binary Pattern (LBP) descriptor and employs sparse projection method to make the high dimensional feature practical. The High Dimensional LBP algorithm achieves 93.18% accuracy under the LFW dataset in unrestricted protocol [7], but its accuracy drops to 63.2% under the multi-views and illuminations dataset [5]. In fact, the problem of pose-invariant in face recognition which is desired by many applications remains unsolved, as argued by Ref. [8]. In their research, a combination of geometric and statistical modeling techniques was used to solve the pose problem in face recognition. The algorithm can reach 99.19% accuracy in near-frontal face recognition (between -15° to 15°) and the accuracy drops significantly to an average of around 75% when the poses change to 45 degrees from frontal images. This experiment shows that the key ability of pose-invariant face recognition has not been solved yet.

Briefly, the process of face recognition can be divided into the following steps: (1) Face detection, where the computer searches the existence of face features in an image or a video. Commonly, the output of face detection is a bounding box around each face. (2) Face alignment, where we align the face image to standardize template by resizing the image size and correcting the location and orientation of the face. (3) Feature extraction, where the computer extracts the derived features of the face image by using representation method. (4) Face identification, where we compare the similarity between the input image and the faces in the database and identify to whom the input image belongs to.

The remainder of this research is organized as follows: the next section will review face detection and face alignment in detail. Furthermore, we discuss feature extraction and face identification in Section III. In Section IV, we implement the multi-pose face recognition based on Principle Component Analysis (PCA). The experiment results of the measurement by using several multi-pose face videos are presented in Section V. Finally, the last section presents the main conclusions of this research.

II. FACE DETECTION AND ALIGNMENT

A. Face Detection

Viola and Jones [9] proposed a face detection algorithm that could achieve real-time face detection with high detection rates. The algorithm was implemented in OpenCV and could detect frontal faces and profile face at about five frames per second. The algorithm

can be divided into four steps: (1) Haar-like Feature Selection. Human faces share several similar characteristics, which can be extracted by using Haar-like (rectangular) features. These features are faster than pixel-based image processing. (2) Calculating Integral Image. The rectangular features are represented by the integral image. By using the integral image, the sum of values in a rectangular subset of the image can be calculated quickly and efficiently. (3) Adaboost Training. AdaBoost which stands for Adaptive Boosting is formulated by Yoav Freund [10]. AdaBoost is used to construct complex features by using only a few simple rectangular features. (4) Cascading Classifiers. The rectangular features as classifiers are arranged in a cascade in order of the complexity. By the cascade structure, the speed of the detection can be increased by focusing only on the most probable areas in the image.

B. Face Alignment

The output of face detection is usually just a rough bounding box around each face. This detected face image has to be aligned in a pre-defined template to compensate the variation of location, size, illumination and orientation. This alignment process aims to get higher accuracy in localizing and normalizing face image because face detection step just provides the rough estimation of location and scale of the face image.

To cope with pose variation problem, we use multi-view face recognition approach by using both frontal face and profile face detection to capture all view of face poses. Therefore, our approach can be considered as extensions of the existing frontal face recognition. Then, we perform some preprocessing steps to the detected faces, which are: (1) the size of face images is set to 100×100 pixels, (2) the image is converted to grayscale value and, (3) the illumination is normalized using histogram equalization algorithm. Figure 1 shows the detected and aligned frontal and profile faces as result of face detection and alignment by using OpenCV.



Fig. 1. The Detected and Aligned Frontal and Profile Faces.

III. FEATURE EXTRACTION AND FACE IDENTIFICATION

After a face image has been aligned and normalized, the feature extraction is done to get the effective input data that robust from geometric and photometric variation [11]. Finally, the face identification is done by conducting feature matching of extracted features for recognizing face images of different peoples. In the identification step, we compare new input face image to the training face images which are saved in face database.

A. Principle Component Analysis

In the feature extraction step, it uses Principal Component Analysis (PCA). The reason for this choice is that natural face images have significant statistical redundancy, which can be reduced to form a more compact representation by using PCA [12]. PCA, which is also known as Karhunen-Loeve Transformation (KLT), gives the orthogonal decomposition of the face image. Its output which is called as eigen image is a linear projection of the input face image corresponding to the largest Eigenvalue of the covariance matrix.

Essentially, the covariance matrix is built from training images that are taken from many objects. To get the covariance matrix, a 2D image with size m -columns and n -rows have to be flattened as a 1D vector. In this research, the size of column and row of the image are the same n dimension. Then, the vector is formed $n^2 \times 1$ dimension. Exactly, all face images in our experiments are 100×100 dimension matrices which are flattened to 10000×1 dimension vectors. Each face image is formed as a 100×100 dimension matrices, as following:

$$I_i = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,100} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,100} \\ \vdots & \vdots & \ddots & \vdots \\ a_{100,1} & a_{100,2} & \cdots & a_{100,100} \end{bmatrix}. \quad (1)$$

The matrix contains the value of the pixel of a 100×100 face image. This matrix is flattened to a vector in 10000×1 dimension space. It can be seen in the matrix below.

$$\Gamma_i = [a_{1,1} \ \cdots \ a_{1,100} \ a_{2,1} \ \cdots \ a_{100,100}]^T \quad (2)$$

If there are N individuals, who become samples, and are taken the P images from, the total images of the training set will be:

$$M = N \times P. \quad (3)$$

The average face image of the training data, as shown in Fig. 2, is computed by using the formula:

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i. \quad (4)$$

After this calculation, the average value of the images is subtracted by each face vector Γ_i to obtain vector Φ_i .

$$\Phi_i = \Gamma_i - \Psi. \quad (5)$$

Finally, the covariance matrix C can be written as: $C = AA^T$, where $A = [\Phi_1 \ \Phi_1 \ \cdots \ \Phi_M]$ and M are the numbers of total images. The matrix A has $M \times 100^2$ dimension, and matrix C has $M \times M$ dimension. By computing the orthogonal decomposition of the matrix C , we will get M eigenvalues and $M \times 100^2$ dimension eigenvectors. Commonly, the number of total images (M) is much less than image size (n^2). This approach is used to reduce the image representation.

If the most significant eigenvector from the previous step is converted back to 100×100 dimension matrix, then it will create face-like images which can be seen in Fig. 3. This is the reason why PCA is also called as eigenface method.

B. Euclidean Distance

If there is a new 100×100 face image Γ that must be recognized, then the following steps should be done:

- 1) The input data Γ is normalized using average face image, as follow:

$$\Omega = \Gamma - \Psi. \quad (6)$$



Fig. 2. Average Face Image.



Fig. 3. Eigenfaces Image

- 2) The normalized images are projected to the Eigenspace. The projection result, which is $M \times 1$ vector is:

$$\Omega^{\text{proj}} = [\Omega^{\text{proj},1} \quad \Omega^{\text{proj},2} \quad \dots \quad \Omega^{\text{proj},M}]^T \quad (7)$$

- 3) To identify the projected image Ω^{proj} , all training images Φ_i need to be projected to eigenspace and become Φ_i^{proj} . Next, to estimate the similarities between two projected images and determine whether two projected images come from the same person, we use the Euclidean distance as similarity measurement. This distance has the formula:

$$\|\Omega^{\text{proj}} - \Phi_i^{\text{proj}}\|_e = \sqrt{\sum_{j=1}^M |\Omega^{\text{proj},j} - \Phi_i^{\text{proj},j}|^2} \quad (8)$$

- 4) After the calculations, the smallest distance value is selected:

$$\Delta = \min_i \|\Omega^{\text{proj}} - \Phi_i^{\text{proj}}\|_e \quad (9)$$

Finally, the conclusion is taken after comparing Δ to the threshold value θ by using the following rules:

- a) If $\Delta < \theta$, where θ is the threshold value that is chosen heuristically, then the input face image is recognized as the same person in a face image in training data with the smallest distance value.
- b) If $\Delta > \theta$, then the input face is not recognized as known person based on face image database.

By using PCA, the computation is reduced from n^2 dimension face image to M dimension projected image. Practically, the total of face images in training set is smaller than the total of pixels in the image [12, p. 96]. Our PCA method approach for face recognition is shown in Fig. 4.

IV. IMPLEMENTATION

The implementation of the algorithm uses .Net framework with C# language programming and EmguCV library. The class HaarCascade in EmguCV is used as the classifier of object detection. To solve pose variation problem, we use both frontal face and profile face detectors to capture all view of face poses. However, the profile face detector is only for the left profile face. To get the right profile face, we flip the image horizontally and perform face detection using the left profile face detector. Therefore, the detection system will be able to detect all view of face poses. These used detectors are referred to the development of face detector research by Seo [13] which build

a decision tree to detect face image. This research resulted in classifier with 95.4% of accuracy rate for the testing dataset.

Our approach can be considered as extensions of the existing frontal face recognition by using Principal Component Analysis method. The only different is that the person faces in the training data are stored in the various pose. In our initial experiments, it is found that the accuracy of identification would decrease if the faces are arranged without regarding the similarities of the poses. To improve the accuracy of the system, the person faces in the training data are separated based on the captured pose, which is face image (1) from the front, (2) from the right side, and (3) from the left side (see Fig. 5).

By splitting into three processes, the face detection and recognition require longer time in sequential processing. If we use this sequential approach, the application cannot run in real time during the face detection and recognition process with the input source using a webcam. Therefore, the implementation of our application uses the concept of parallel processing provided by .NET Framework 4. The main benefit of the parallel processing is the distribution of the data processing tasks to multiple cores in the processor (CPU). Thus, many tasks can be completed more quickly. The usage of the parallel processing will depend on the specification of the CPU. Figure 6 is the example of detected result of left profile face by using our application.

V. EXPERIMENTAL RESULTS

For our experiments, we need to collect our dataset because there is no public face dataset which is directly related to the pose variations problem. Our dataset is designed to answer the experiment objective that is to find the optimal number of variations in the database

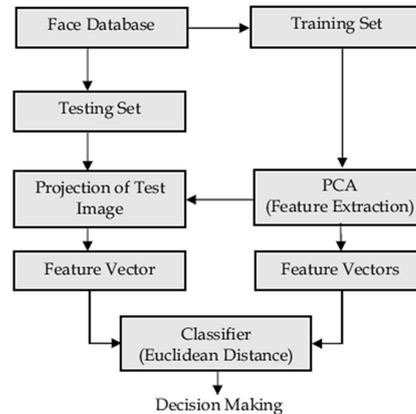


Fig. 4. PCA Approach for Face Recognition [12, p. 96].

and their quantitative pose positions. The collection of face images used in the experiment is captured with a controlled background by using a huge white paper in the room condition that has a light intensity of approximately between 150 to 300 lux. Furthermore, each facial image whether as training data or input data is grayscale and histogram equalized to obtain images with similar gray level value. The angle taken for the facial images is based on yaw head movement. The person faces in the training data are captured by the rule of yaw angle, between -75° up to 75° , as shown in Fig. 7. The total of captured face images is eleven images for each person. The angle measurement is measured using a piece of paper with angle lines placed on the center point of the head.

For the first experiment, the face images in training data are taken from ten persons that each of them

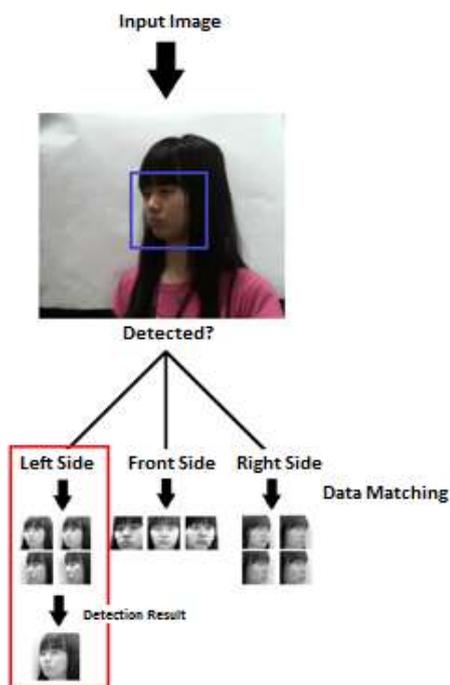


Fig. 5. Process of Face Detection and Recognition with Various Poses

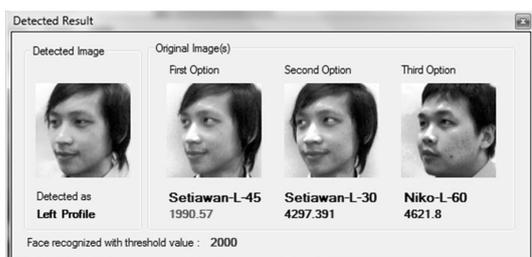


Fig. 6. Example of Detected Result.

has eleven images. The experiment would like to see the recognition system accuracy without tuning threshold parameter. It is done by setting a threshold value equal to -1 , where all detected faces will return the recognition results. In this experiment, we get the person identification of the smallest distance Δ . Finally, the experiment result is shown in Table I. In the Table, the term 'recognizable' means the detected faces, and 'amount of recognizable' means the number of the detected faces. Furthermore, the term 'amount of successful' means the number of successful recognition of the detected faces.

Based on the experiment result above, the accuracy of the system in recognizing every detected face can be calculated as:

$$\text{Accuracy} = \frac{\sum \text{Successful}}{\sum \text{Recognizable}} \times 100 = 69.91\% \quad (10)$$

The second experiment would like to see the effect of pose variation in training data by reducing certain poses in the first experiment. First, we create five variations based on a number of face image pose in training data as shown in Table II.

The impact of training data variation based on the detected pose angles in the recognition accuracy is shown in Table III. The result of last variation (V) is the same as the first experiment.

In the third experiment, we would like to analyze the impact of varying threshold value to the successful recognition. The variation of the threshold is from 1000 to 5000, and this threshold is compared to the smallest



Fig. 7. All view of the face poses.

TABLE I
EXPERIMENT RESULT WHEN SYSTEM WITHOUT THRESHOLD.

| No | Sample | Amount of Recognizable | Amount of Successful |
|-------|--------|------------------------|----------------------|
| 1 | A | 158 | 89 |
| 2 | B | 201 | 130 |
| 3 | C | 214 | 156 |
| 4 | D | 147 | 102 |
| 5 | E | 101 | 74 |
| 6 | F | 71 | 38 |
| 7 | G | 158 | 133 |
| 8 | H | 64 | 62 |
| 9 | I | 185 | 138 |
| 10 | J | 150 | 91 |
| Total | | 1449 | 1013 |

TABLE II
TRAINING DATA VARIATION.

| Variation (#Images) | Angle variation (degree) | | |
|------------------------|--------------------------|------------|----------------|
| | Left Side | Front Side | Right Side |
| I (3) | 45 | 0 | 45 |
| II (5) | 30, 60 | 0 | 30, 60 |
| III (6) | 30, 60 | 0 | 30, 60 |
| IV (8) | 45, 60, 75 | -15, 15 | 45, 60, 75 |
| V (11) | 30, 45, 60, 75 | -15, 0, 15 | 30, 45, 60, 75 |

TABLE III
EXPERIMENT RESULT BASED ON TRAINING DATA VARIATION.

| Variation | Amount of Recognizable | Amount of Successful | Accuracy |
|-----------|------------------------|----------------------|----------|
| I | 1396 | 650 | 46.56 |
| II | 1450 | 848 | 58.48 |
| III | 1472 | 950 | 64.54 |
| IV | 1451 | 917 | 63.20 |
| V | 1449 | 1013 | 69.91 |

TABLE IV
SIMULATION RESULT BASED ON THRESHOLD VALUE VARIATION.

| No | Maximum Threshold Value | Amount of Recognizable under Threshold Value | Amount of Successful | Accuracy |
|----|-------------------------|--|----------------------|----------|
| 1 | 5000 | 1417 | 1006 | 0.71 |
| 2 | 4000 | 1276 | 944 | 0.74 |
| 3 | 3000 | 926 | 743 | 0.80 |
| 4 | 2000 | 337 | 332 | 0.99 |
| 5 | 1000 | 15 | 15 | 1.00 |

distance as described in Section III. Furthermore, this experiment uses only the variation V in Table II, and the experiment result of recognition accuracy based on threshold value is shown in Table IV. In this table, the term 'amount of recognizable under threshold value' means the number of the detected faces if the recognition system is set to the certain threshold value. The threshold value represents the recognition accuracy. The lower the threshold value is, the higher the recognition accuracy will be.

The last experiment is to know the processing time in loading the variation of face image training data. The experiment result is shown in Table V.

VI. CONCLUSIONS

The failure in recognizing input face based on our visual observation could be caused by the digital image

TABLE V
PROCESSING TIME IN LOADING TRAINING DATA

| Variation | Face Load Time (ms) | | |
|-----------|---------------------|------|-------|
| | Front | Left | Right |
| I | 17 | 9 | 9 |
| II | 19 | 33 | 33 |
| III | 43 | 32 | 32 |
| IV | 42 | 72 | 72 |
| V | 81 | 126 | 128 |

which is not clear or has excessive noise, the close similarity between one face with another, and the pose with the certain angle which is not provided in training data. These three reasons have the main contribution in decreasing recognition accuracy. Based on the analysis of the experiment results in Tables I, III–V, our conclusions about recognition performance in facing pose variation are as following:

- 1) The more pose variation of face images used as training data is, the better recognition results are, but it also increases the processing time.
- 2) The lower threshold value is, the harder it is to recognize a face image, but it also increases the accuracy. This setting is suitable for authentication application.
- 3) The higher threshold value is, the easier it is to recognize a face image, but it also reduces the accuracy. This setting is suitable for surveillance application.
- 4) The best result of our experiment is 100% of recognition accuracy rate for the threshold value of 1000, and the worst result is 69.91% of recognition accuracy rate when a threshold value equals to -1. This threshold means that all detected faces will return the recognition results.

The main weakness in our approach is that there is no classification algorithm after reducing the dimension using PCA. Therefore, the additional classification algorithm will be the first issue in our next step research, and it will be addressed using algorithm establishing the classification of multiple data, such as Support Vector Machines (SVM).

REFERENCES

- [1] A. T. Tokuhiko and B. J. Vaughn, "Initial test results of the omron face cue entry system at the university of missouri-rolla reactor," *Journal of nuclear science and technology*, vol. 41, no. 4, pp. 502–510, 2004.
- [2] E. Lococo. (2002, May) Airport switches face-recognition systems over accuracy con-

- cerns. Bloomberg News. [Online]. Available: <http://articles.latimes.com/2002/may/31/business/fi-face31>
- [3] Sujono and A. A. Gunawan, "Face expression detection on kinect using active appearance model and fuzzy logic," *Procedia Computer Science*, vol. 59, pp. 268–274, 2015.
- [4] A. A. S. Gunawan and H. Setiadi, "Handling illumination variation in face recognition using multiscale retinex," in *Proceedings of International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Malang, Indonesia, 2016.
- [5] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Multi-view perceptron: a deep model for learning face identity and view representations," in *Advances in Neural Information Processing Systems*, 2014, pp. 217–225.
- [6] E. Learned-Miller, G. B. Huang, A. Roy-Chowdhury, H. Li, and G. Hua, *Labeled Faces in the Wild: A Survey*. Cham: Springer International Publishing, 2016, pp. 189–248. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-25958-1_8
- [7] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3025–3032.
- [8] R. Abiantun, U. Prabhu, and M. Savvides, "Sparse feature extraction for pose-tolerant face recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 10, pp. 2061–2073, 2014.
- [9] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.
- [10] Y. Freund, "Boosting a weak learning algorithm by majority," *Information and Computation*, vol. 121, no. 2, pp. 256–285, 1995.
- [11] S. Z. Li and A. K. Jain, *Handbook of Face Recognition*. New York: Springer Science, 2005.
- [12] A. Eleyan and H. Demirel, *Pca and lda based neural networks for human face recognition*. INTECH Open Access Publisher, 2007.
- [13] N. Seo. (2008, October) Tutorial: Opencv haar training (rapid object detection with a cascade of boosted classifiers based on haar-like features). [Online]. Available: <http://note.sonots.com/SciSoftware/haartraining.html>.