

Modeling of the Gaussian-Based Component Analysis on the Kernel Space to Extract Face Image

Arif Muntasa^(✉) and Indah Agustien Siradjuddin

Informatics Engineering Department, Engineering Faculty, University of Trunojoyo, Ry Telang Po. Box 2 Kamal, Madura, Bangkalan, Indonesia
arifmuntasa@trunojoyo.ac.id

Abstract. One of the methods to extract the image characteristics on the feature space is Kernel Principal Component Analysis. Gaussian is a model to transform the image to the features spaces by using kernel trick. In this paper, the new model is proposed to add the image features to be more dominant, so that the main image features can be raised. Two databases were used to verify the proposed method, which are the YALE and the CAI-UTM. Three scenarios have been applied with different training samples. The results demonstrated that the proposed method can recognize the face image 87.41% for two training sets, 90.83% for three training sets, and 92.38% for four training sets on the YALE database. On the CAI-UTM database, the proposed method could classify correctly by 83.75%, 85.57%, and 87.33% for two, three, and four training sets respectively. The comparison results show that the results of the proposed approach outperformed to other methods.

Keywords: Gaussian kernel · Component analysis · Feature selection

1 Introduction

Image analysis has been developed to solve and aid live problem, one of them is a biometric field. Many algorithms were created and developed by researchers to obtain better results i.e. Eigenface and its extension [1–3], Fisherface [4–7], Laplacianfaces [8–10], Factor analysis, Independent Component Analysis [11], metric multidimensional scaling [12], Isomap [13], Hessian, Laplacian Eigenmaps [14], and other methods.

Eigenface is the simplest method to extract the features of the image by reducing the dimensionality of the image. The Eigenface can reduce the dimensionality of the image up to some data used for the training sets. The Eigenface is obtained from the projection of the eigenvector and the training sets, whereas the eigenvector is derived from the covariance matrix of the training sets. Eigenface can reduce the dimension of the image up to the size of the image dimension minus the amount of data, when the covariance matrix is based-orthogonal. However, the Eigenface can only map the feature linearly, since it can only solve linear model in high dimensionality, while non-linear distribution cannot be solved by the Eigenface. In fact, the used data

distribution cannot be predicted whether linear or non-linear [8, 15]. Besides, the produced features of the Eigenface depend on the amount of training sets utilized. If the number of training sets is more than the image dimensionality, then reduction dimensionality cannot be applied to obtain the features.

Fisherface has improved the weakness of the Eigenface methods, where the features can be reduced so that the number of features to become as the amount of classes. Unlike the Eigenface, Fisherface can distinguish the information, though it is not based-orthogonal, therefore the Eigenface is not better than the Fisherface. However, the Fisherface has also the weakness, i.e. inability to separate non-linear distributed features [4]. The Fisherface will also fail to obtain the features, when the within-scatter class is a singular matrix.

One of the methods that can solve non-linear data distribution is Isomap. It is one graph based-method that used to reduce the dimensionality. However, an Isomap produces a graph with topological instability, though the methods have been improved by reducing several data points on the graph. Another weakness of an Isomap is non-convex manifold because it will succeed the process is not complete.

Another non-linear method is Local Linear Embedding (LLE). It is a graphical method that similar to Isomap method, where Isomap tries to preserve the characteristics of the local data. The weakness of non-convex of the Isomap can be solved by preserving of the local data characteristics. However, several researches have reported that the LLE cannot success to visualize the data points. LLE has delivered the constraints of the data points so that they spread to undesired areas [16].

Two graphical based-methods has also weakness, therefore in this research, we proposed another approach to reduce the image dimensionality. A transformation from the image to the feature spaces is proposed to project the eigenvector of the Gaussian kernel matrix. Gaussian kernel found the distance between the points to the others of the training sets. Gaussian considers standard deviation as inner scale to record the data distribution deviation, while Gaussian kernel is a way to map image to feature spaces by Gaussian equation. The principal component is just applied the Gaussian kernel matrix to obtain the Eigenvalue and Eigenvector. In this case, the average, zero mean, and covariance matrixes are not necessary to be calculated, because the image samples have been mapped into feature spaces by Gaussian Kernel. The results of the Eigenvector are delivered to calculate the projection matrix as the image features. The image features will be further process to classify the face image.

2 Proposed Approach

Kernel trick is method to convert from image to feature spaces. On the feature spaces, there are four models to conduct kernel trick, which are Linear, Gaussian, Polynomial, and Polyplus [15, 17, 18]. Gaussian model is one of the kernel trick model considered the distance, mean, and variant of an object. Therefore, Gaussian kernel trick model can extract the more dominant features than the others, i.e. Linear, Polynomial, and Polyplus. Suppose the training and the testing sets are represented by X and Y , where X has m samples and n image dimensionalities, while Y is row matrix with n columns as described in the Eq. (1).

$$\mathcal{X} = \begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & x_{2,3} & \cdots & x_{2,n} \\ x_{3,1} & x_{3,2} & x_{3,3} & \cdots & x_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & x_{m,3} & \cdots & x_{m,n} \end{pmatrix} \quad (1)$$

For each testing set is written as follows,

$$\mathcal{Y} = (y_{1,1} \quad y_{1,2} \quad y_{1,3} \quad \cdots \quad y_{1,n}) \quad (2)$$

2.1 Gaussian Kernel Matrix

Gaussian is one of models used to map from image to feature spaces [8]. Gaussian-based component analysis on the kernel can be obtained by calculation $K_x(X, X^T)$ and $K_y(Y, Y^T)$, i.e. Gaussian kernel matrix for the training and testing sets. To obtain Gaussian kernel matrix $K_x(X, X^T)$, the distance between point and others is calculated as follows,

$$A_{i,1} = \sum_{j=1}^n (\mathcal{X}_{i,j} \cdot \mathcal{X}_{i,j}) \quad (3)$$

The value of i has range as follows: $i \in 1, 2, 3, \dots, m$. If the Eq. (3) is calculated from $i = 1$ until $i = m$, then the Eq. (5) can be also written as follows,

$$A_{m,1} = \begin{pmatrix} \sum_{j=1}^n (\mathcal{X}_{1,j} \cdot \mathcal{X}_{1,j}) \\ \sum_{j=1}^n (\mathcal{X}_{2,j} \cdot \mathcal{X}_{2,j}) \\ \vdots \\ \sum_{j=1}^n (\mathcal{X}_{m,j} \cdot \mathcal{X}_{m,j}) \end{pmatrix} \quad (4)$$

Furthermore, Eq. (4) is duplicated to be m columns, so that the matrix dimensionality will be m rows and m columns. The effect of column duplication is the same values for each column as shown in Eq. (5)

$$A_{m,m} = \begin{pmatrix} \sum_{j=1}^n (\mathcal{X}_{1,j} \cdot \mathcal{X}_{1,j}) & \sum_{j=1}^n (\mathcal{X}_{1,j} \cdot \mathcal{X}_{1,j}) & \cdots & \sum_{j=1}^n (\mathcal{X}_{1,j} \cdot \mathcal{X}_{1,j}) \\ \sum_{j=1}^n (\mathcal{X}_{2,j} \cdot \mathcal{X}_{2,j}) & \sum_{j=1}^n (\mathcal{X}_{2,j} \cdot \mathcal{X}_{2,j}) & \cdots & \sum_{j=1}^n (\mathcal{X}_{2,j} \cdot \mathcal{X}_{2,j}) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{j=1}^n (\mathcal{X}_{m,j} \cdot \mathcal{X}_{m,j}) & \sum_{j=1}^n (\mathcal{X}_{m,j} \cdot \mathcal{X}_{m,j}) & \cdots & \sum_{j=1}^n (\mathcal{X}_{m,j} \cdot \mathcal{X}_{m,j}) \end{pmatrix} \quad (5)$$

Equation (4) is duplicated to be m columns

$$\mathcal{B}_{m,m} = \left(\begin{array}{cccc} \sum_{j=1}^n (\mathcal{X}_{1,j} \cdot \mathcal{X}_{1,j}) & \sum_{j=1}^n (\mathcal{X}_{1,j} \cdot \mathcal{X}_{1,j}) & \cdots & \sum_{j=1}^n (\mathcal{X}_{1,j} \cdot \mathcal{X}_{1,j}) \\ \sum_{j=1}^n (\mathcal{X}_{2,j} \cdot \mathcal{X}_{2,j}) & \sum_{j=1}^n (\mathcal{X}_{2,j} \cdot \mathcal{X}_{2,j}) & \cdots & \sum_{j=1}^n (\mathcal{X}_{2,j} \cdot \mathcal{X}_{2,j}) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{j=1}^n (\mathcal{X}_{m,j} \cdot \mathcal{X}_{m,j}) & \sum_{j=1}^n (\mathcal{X}_{m,j} \cdot \mathcal{X}_{m,j}) & \cdots & \sum_{j=1}^n (\mathcal{X}_{m,j} \cdot \mathcal{X}_{m,j}) \end{array} \right)^T \Bigg\} m$$

The transpose result of Equation (5)

(6)

Moreover, the Eq. (5) is transposed, and the transpose result is represented by using \mathcal{B} as represented in Eq. (7).

The distance of the training sets can be computed by using the simple operation as shown in Eq. (8), while the value of \mathcal{C} can be calculated by using the Eq. (9)

$$\mathcal{D}_{m,m} = |\mathcal{A}_{m,m} + \mathcal{B}_{m,m} - \mathcal{C}_{m,m}| \quad (7)$$

$$\mathcal{C} = 2 \times \mathcal{X} \times \mathcal{X}^T \quad (8)$$

Based on the Eq. (8), the Gaussian kernel matrix can be obtained by using the Eq. (10), while the variable of σ represents the standard deviation. The value of standard deviation can be defined as positive integer, but usually the values of standard deviation utilized are 1, 2, or 3.

$$\mathcal{K}(\mathcal{X}, \mathcal{X}^T) = \exp\left(-\frac{\mathcal{D}_{m,m}}{2 \times \sigma^2}\right) \quad (9)$$

2.2 Sharpen of Gaussian-Based Component Analysis

In order to calculate the kernel on the feature space (\mathcal{G}), the matrix \mathbb{I} is required, where all of elements are 1. Moreover, the feature space can be processed by the simple operation as written in the following equation

$$\mathcal{G} = \mathcal{K} - (\mathbb{I} \times \mathcal{K}) - (\mathcal{K} \times \mathbb{I}) + (\mathbb{I} \times \mathcal{K} \times \mathbb{I}) + (\mathcal{K} \times \mathbb{I} \times \mathcal{K}) \quad (10)$$

The operation of $(\mathcal{K} \times \mathbb{I} \times \mathcal{K})$ is utilized to sharpen the features of the object, so that the object is easier to be recognized. The addition of the operation in the Eq. (12) has indicated that the dominant features can be maximally extracted. The result of Eq. (12) is furthermore applied to gain the eigenvalues and eigenvectors as written in Eq. (12)

$$\text{Det}(\mathcal{G} - I \times \lambda) = 0 \quad (11)$$

The variable of I represents the identity matrix which is the matrix with zero elements except the main diagonal with 1 value. The calculation result of the Eq. (13) produces the eigenvalues and eigenvectors as shown in Eqs. (13) and (14)

$$\lambda = \begin{pmatrix} \lambda_{1,1} & 0 & 0 & \cdots & 0 \\ 0 & \lambda_{2,2} & 0 & \cdots & 0 \\ 0 & 0 & \lambda_{3,3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_{m,m} \end{pmatrix} \quad (12)$$

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} & A_{1,3} & \cdots & A_{1,m} \\ A_{1,2} & A_{2,2} & A_{2,2} & \cdots & A_{2,m} \\ A_{1,3} & A_{3,2} & A_{3,3} & \cdots & A_{3,m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{1,m} & A_{m,2} & A_{m,3} & \cdots & A_{m,m} \end{pmatrix} \quad (13)$$

The eigenvalues as represented in Eq. (13) can be also represented by using row matrix as shown in Eq. (15).

$$\lambda = (\lambda_{1,1} \quad \lambda_{2,2} \quad \lambda_{3,3} \quad \cdots \quad \lambda_{m,m}) \quad (14)$$

These values of λ are not decreasingly ordered yet, therefore these values must be decreasingly ordered. The sorting result is represented by using $(\hat{\lambda})$ as shown in the following equation,

$$\hat{\lambda}_{1,1} \geq \hat{\lambda}_{2,2} \geq \hat{\lambda}_{3,3} \geq \hat{\lambda}_{4,4} \geq \cdots \hat{\lambda}_{m,m} \quad (15)$$

The change of column position is also effect of column position of the eigenvectors. The sorting results of the eigenvectors (\hat{A}) are composed based on the index found of the sorting results of the eigenvalues as written in Eq. (17).

$$\hat{A} = \begin{pmatrix} \hat{A}_{1,1} & \hat{A}_{1,2} & \hat{A}_{1,3} & \cdots & \hat{A}_{1,m} \\ \hat{A}_{1,2} & \hat{A}_{2,2} & \hat{A}_{2,2} & \cdots & \hat{A}_{2,m} \\ \hat{A}_{1,3} & \hat{A}_{3,2} & \hat{A}_{3,3} & \cdots & \hat{A}_{3,m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{A}_{1,m} & \hat{A}_{m,2} & \hat{A}_{m,3} & \cdots & \hat{A}_{m,m} \end{pmatrix} \quad (16)$$

2.3 Projection of the Gaussian-Based Component Analysis

To obtain the features using Gaussian kernel, for both the training \mathcal{P}_x and testing Gaussian kernel \mathcal{P}_y can be simply represented.

$$\mathcal{P}_x = \mathcal{K}_x(\mathcal{X}, \mathcal{X}^T) \times \hat{A} \quad (17)$$

In this case, $\mathcal{K}_x(\mathcal{X}, \mathcal{X}^T)$ is the Gaussian kernel matrix for the training sets, it was obtained by using Eq. (10), while the matrix values $\hat{\lambda}$ was also calculated by using Eq. (17).

$$\mathcal{P}_y = \mathcal{K}_y(\mathcal{Y}, \mathcal{Y}^T) \times \hat{\lambda} \quad (18)$$

$\mathcal{K}_y(\mathcal{Y}, \mathcal{Y}^T)$ is the Gaussian kernel matrix for the testing sets. The difference between $\mathcal{K}_x(\mathcal{X}, \mathcal{X}^T)$ and $\mathcal{K}_y(\mathcal{Y}, \mathcal{Y}^T)$ is used as the input. $\mathcal{K}_x(\mathcal{X}, \mathcal{X}^T)$ applies training sets as the input (see Eq. (1)), whereas $\mathcal{K}_y(\mathcal{Y}, \mathcal{Y}^T)$ applies the testing sets as input (see Eq. (2)).

The calculation result of Eqs. (18) and (19) can be written in the following matrix as seen in Eq. (20) for \mathcal{P}_x and in Eq. (21) for \mathcal{P}_y .

$$\mathcal{P}_x = \left(\begin{array}{ccccc} P_{1,1} & P_{1,2} & P_{1,3} & \cdots & P_{1,m} \\ P_{1,2} & P_{2,2} & P_{2,2} & \cdots & P_{2,m} \\ P_{1,3} & P_{3,2} & P_{3,3} & \cdots & P_{3,m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_{1,m} & P_{m,2} & P_{m,3} & \cdots & P_{m,m} \end{array} \right) \Bigg\} m \quad (19)$$

$$\mathcal{P}_y = (\hat{P}_{1,1} \quad \hat{P}_{1,2} \quad \hat{P}_{1,3} \quad \cdots \quad \hat{P}_{1,m}) \quad (20)$$

2.4 Features Selection and Similarity Measurements

As mentioned in Eqs. (20) and (21), the column dimensionality of them is m . It is indicated that the number of features produced is m , for both the training and the testing sets. However, features generated will not be used all for measurement. Therefore the features produced must be selected to be applied on similarity measurements. The selection results can be shown in Eqs. (22) and (23).

$$\mathcal{P}_x = \left(\begin{array}{cccc|ccc} P_{1,1} & P_{1,2} & \cdots & P_{1,t} & P_{1,t+1} & \cdots & P_{1,m} \\ P_{2,1} & P_{2,2} & \cdots & P_{2,t} & P_{2,t+1} & \cdots & P_{2,m} \\ P_{3,1} & P_{3,2} & \cdots & P_{3,t} & P_{3,t+1} & \cdots & P_{3,m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ P_{m,1} & P_{m,2} & \cdots & P_{m,t} & P_{m,t+1} & \cdots & P_{m,m} \end{array} \right) \quad (21)$$

t training features are used $(m-t)$ features are not used

$$\mathcal{P}_y = \left(\underbrace{\hat{P}_{1,1} \quad \hat{P}_{1,2} \quad \cdots \quad \hat{P}_{1,t}}_{t \text{ testing features are used}} \left| \underbrace{\hat{P}_{1,t+1} \quad \hat{P}_{1,t+2} \quad \cdots \quad \hat{P}_{1,m}}_{(m-t) \text{ features are not used}} \right. \right) \quad (22)$$

The first column represented the more dominant feature than the second, the third, until the m^{th} columns. It means that the bigger the column index, the feature less dominant. The feature selection is also intended to reduce the computation time when the similarity measurements are applied to classify the face images. In order to classify the face image, the simple method is applied, which is the city block method as shown in Eq. (23)

$$\mathcal{D}_{m,1} = |\mathcal{P}_x - \mathcal{P}_y| \quad (23)$$

3 Experimental and Discussion

In order to evaluate the proposed approach, two face databases have been prepared, which are the YALE and the CAI-UTM database. The YALE face database is a small database, which has only fifteen people, where for each people has eleven different poses [19], sample of face image can be seen in see Fig. 1. Though, it is small face database, but the images have many illumination, accessories, and expressions variation. The second face database is the CAI-UTM, where it has a hundred people, and for each people has ten different poses as shown in Fig. 2 as a sample of image. Therefore, the second face database has a thousand images.



Fig. 1. The yale sample



Fig. 2. CAI-UTM sample

3.1 The Results on the Yale Database

In this paper, three scenarios are implemented to evaluate the proposed approach, which are using two, three, and four face images as the training sets. Each scenario will be conducted five times experiments with different images indexes (five-fold cross

validation). For each scenario, the features will be selected based on the training sets applied. For the first scenario, 20 until 29 features are selected, 30 until 39 features for the second scenario, and 40–49 features for the last scenario.

The results of the first scenario can be seen in Fig. 3. The accuracy of each experiment and the average accuracy were described in the figure. The investigation results show that the accuracy depends on the images used as the training sets. The use of face image training with illumination and expression can produce higher accuracy than the others. In this case, the second experiment has delivered the highest accuracy. The use of features also influences the accuracy produced. Based on Fig. 3, the more features applied, the higher accuracy delivered. It is also shown on the average accuracy obtained, where the line of accuracy tend increase proportional to number of features applied. The results of the first scenario, maximum accuracy produced is 87.41%, while the average accuracy is 80.54%.

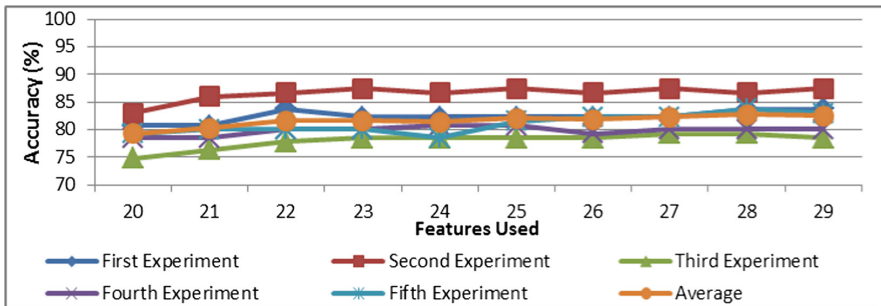


Fig. 3. The first scenario on the YALE

The similar results are also shown in the second scenario, where the use of the illumination and expression face images as the training sets has delivered the better results than the others. On the second scenario has shown the best performance, it is influenced by the sample images used, which is illumination and expression images has been applied as the training sets. The worst result has occurred on the fourth experiment, the investigation results show that the use of normal images as training sets will delivered the lower accuracy than the illumination and expression face images as the training sets, even the more features applied, the less the accuracy obtained.

Based on Fig. 4, the maximum accuracy is 90.83%, the results show that the third scenario is better than the second scenario. This can also be seen from the obtained average accuracy, the third scenario has delivered higher average accuracy than in the second scenario, which is 87.22%.

Four images as the training sets are applied on the last scenario, the worst result (more than 84%) is still better than the average accuracy of the first scenario (less than 81%). The average accuracy of the last scenario (more than 88%) is still better than the maximum accuracy of the first scenario (less than 88%). Similar to the third scenario, the fourth experiment has delivered the worst result on the last scenario, it is caused by the sample used has not been representative of the training sets. The result of the last

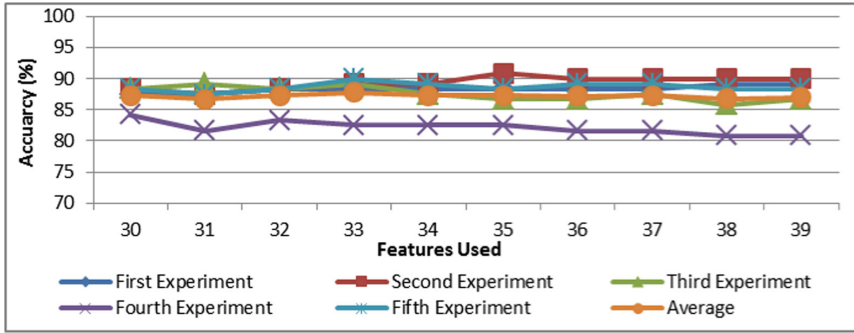


Fig. 4. The second scenario on the YALE

scenario can be shown in Fig. (5). The average accuracy also tends to increase in proportion to number features used. Maximum and average accuracies achieved are 92.38% and 88.91%.

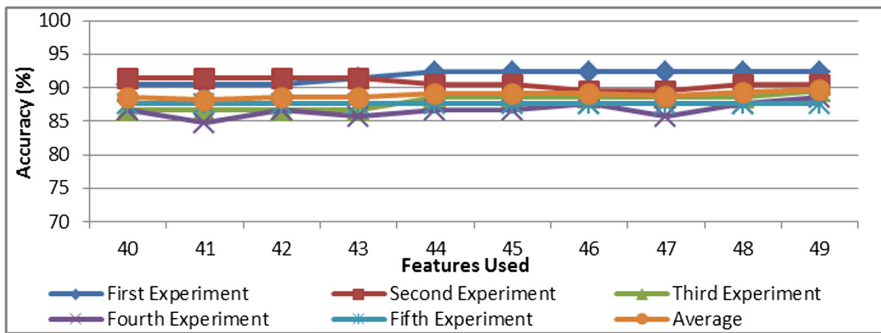


Fig. 5. The third scenario on the YALE

The results were also compared to other approaches for all of scenarios, which are Eigenface, Fisherface, and Laplacianfaces. The performance results of the first and the second scenarios show that the proposed approach outperformed to other methods, while the last scenario shows that Fisherface and Laplacianfaces are better than the proposed approach for accuracy average, but the proposed approach outperformed to the others for the maximum accuracy as seen in Fig. 6.

3.2 The Results on the CAI-UTM Database

Different database was also applied to evaluate the performance of the proposed approach, which is the CAI-UTM. A thousand images are prepared from a hundred people [20]. The Difference with the previous database, in this face database, the number of used features is more than previous database. Since the number of captured sample people is more. For the first scenario, 185 until 199 features are selected, 285

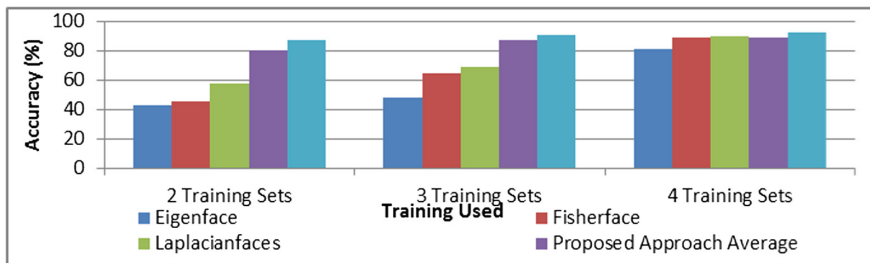


Fig. 6. Performance of the proposed approach was compared to other methods

until 299 features for the second scenario, and 385–399 features for the last scenario. Each scenario is tested using 5-fold Cross Validation.

The first scenario results can be displayed in Fig. 7. Six different line models were described in Fig. 7. However, four line models are almost overlapping, because they produce similar similarity for each feature, include the average accuracy line. As shown in Fig. 6, the proposed approach delivered the stable accuracy result for each features.

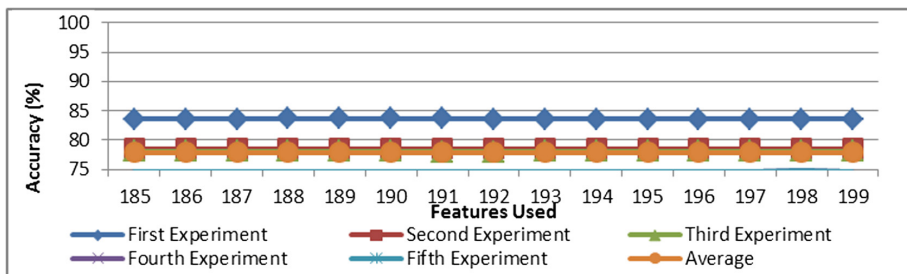


Fig. 7. The first scenario on the CAI-UTM

The best performance has occurred on the first experiment, while the worst results are in the last experiment. By using representative face image samples, the proposed approach system can recognize face image model. Representative face image samples are face image with different models, i.e. open smile, close smile or other expressions. But if the samples used are normal face without expressions, then the proposed approach sometimes cannot recognize the face with other expressions, such as surprise or the others. Performance maximum and average of the proposed approach are 83.75% and 77.90%.

The second scenario is evaluated by using three image samples. The obtained results are shown in Fig. 8. The result shows that the fewer features have represented the characteristics of an image. The second and fifth experiments have delivered the highest performance, because they have applied the image with different poses that represent the other poses, while the other experiments used the images with similar poses, therefore the produced features do not represent overall image poses on the same class. The maximum (85.57%) and average (84.01%) performance of the second

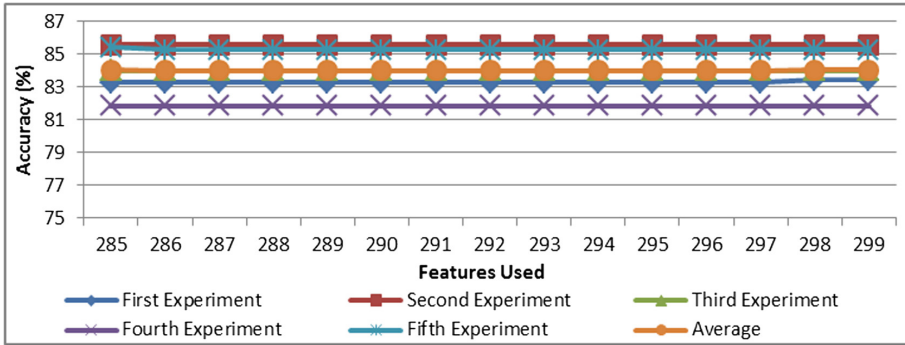


Fig. 8. The second scenario on the CAI-UTM

experimental results is better than the first experiment results, it indicated that the accuracy is affected by number of training samples and poses applied.

The results of the last scenario show the similar trend to the first and the second scenario, i.e. the number of features clearly influenced the performance of the proposed approach. The final scenario results are also evident that the use of training sets with diverse samples is able to produce better accuracy than similar poses as described in Fig. 9. The results of the use of diverse training sets can be seen in the third and fifth experiments, whereas the similar use of the images as the training sets can be seen in the first, second and fourth experiments. The experimental results also proved that the use of the amount of training data affects the resulting accuracy, which are 87.33% for the maximum and 86.43% for the average.

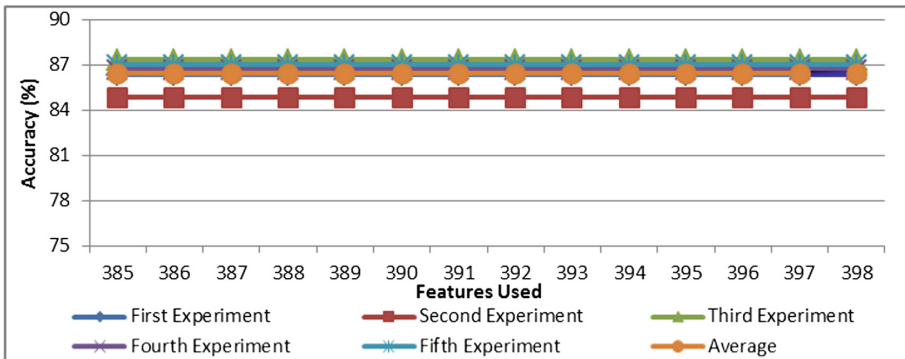


Fig. 9. The last scenario on the CAI-UTM

4 Conclusion

In this paper, the proposed approach has proved that modeling of the Gaussian-based Component Analysis on the kernel space can extract the image features by reducing the dimensionality of the training sets. The proposed approach can recognize the face image

with small sample training sets and even it is better than other methods, i.e. Eigenface, Fisherface, and Laplacianfaces. The proposed approach was also evaluated by using local face image database, i.e. CAI-UTM database, where the evaluation results show that the proposed approach is able to recognize facial image more than 87%.

Acknowledgement. This research has been covered by Ministry of Research Technology and Higher Education and facilitated by Laboratory of Computational Artificial Intelligence, University of Trunojoyo Madura, 2017.

References

1. Zou, H., Hastie, T., Tibshirani, R.: Sparse principal component analysis. *J. Comput. Graph. Stat.* **15**(2), 265–286 (2006)
2. Lu, H., Plataniotis, A.N.V.K.N.: MPCA: multilinear principal component analysis of tensor objects. *IEEE Trans. Neural Netw.* **19**(1), 18–39 (2008)
3. Huang, S.M., Yang, J.F.: Improved principal component regression for face recognition under illumination variations. *IEEE Trans. Sig. Process. Lett.* **19**(4), 179–182 (2012)
4. Muntasa, A.: New modelling of modified two dimensional fisherface based feature extraction. *Telkomnika (Telecommun. Comput. Electron. Control)* **12**(1), 115–122 (2014)
5. Muntasa, A., Siradjuddin, I.A., Wahyuningrum, R.T.: Multi-criteria in discriminant analysis to find the dominant features. *Telkomnika* **14**(3), 1113–1122 (2016)
6. Ye, J., Li, Q.: A two-stage linear discriminant analysis via QR-decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(6), 929–941 (2005)
7. Zheng, W.-S., Lai, J.H., Li, S.Z.: 1D-LDA vs. 2D-LDA: when is vector-based linear discriminant analysis better than matrix-based? *Pattern Recogn.* **41**(7), 2156–2172 (2008)
8. Muntasa, A.: The Gaussian orthogonal laplacianfaces modelling in feature space for facial image recognition. *Makara J. Technol.* **18**(2), 79–85 (2014)
9. Niu, B., Yang, Q., Shiu, S.C.K., Pal, S.K.: Two-dimensional Laplacianfaces method for face recognition. *Pattern Recogn.* **41**(10), 3237–3243 (2008)
10. He, X., Yan, S., Hu, Y., Niyogi, P., Zhang, H.-J.: Face recognition using Laplacianfaces. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(3), 328–340 (2005)
11. Bartlett, M.S., Movellan, J.R., Sejnowski, T.J.: Face recognition by independent component analysis. *IEEE Trans. Neural Netw.* **13**(6), 1450–1464 (2002)
12. Biswas, S., Bowyer, K.W., Flynn, P.J.: Multidimensional scaling for matching low-resolution face images. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(10), 2019–2030 (2012)
13. Yang, M.-H.: Face recognition using extended isomap. In: 2002 International Conference on Image Processing on Proceedings, Rochester, NY, USA. IEEE (2002)
14. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.* **15**(6), 1373–1396 (2003)
15. Muntasa, A.: The human facial expression classification using the center Kernel subspace based the ridge regression. *J. Comput. Sci.* **11**(11), 1054–1059 (2015)
16. Memisevic, R., Hinton, G.: Improving dimensionality reduction with spectral gradient descent. *Neural Netw.* **18**(1), 702–710 (2005)
17. Pumomo, M.H., Sarjono, T.A., Muntasa, A.: Smile stages classification based on Kernel Laplacian-lips using selection of non linear function maximum value. In: VECIMS 2010 - 2010 IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems on Proceedings, pp. 151–156. IEEE, Taranto-Italy (2010)

18. Nadil, M., Souami, F., Labed, A., Sahbi, H.: KCCA-based technique for profile face identification. *EURASIP J. Image Video Process.* **2017**(2), 1–13 (2017)
19. Yale Face Image Database. <http://www.face-rec.org/databases/>. Accessed 2000
20. CAI-UTM, CAI-UTM Face Image Database (2016)