



Comparison of Geometric Features and Color Features for Face Recognition

Cahya Rahmad^{1*} Kohei Arai² Rosa A. Asmara¹ Ekojono Ekojono¹
 Dimas R. H. Putra³

¹Information Technology Department, State Polytechnic of Malang, Indonesia

²Graduate School of Science and Engineering, Saga University, Japan

³Department of Electronic Engineering, State Polytechnic of Malang, Indonesia

* Corresponding author's Email: cahya.rahmad@polinema.ac.id

Abstract: Face recognition plays an important role in the identity recognition system, the color and geometry feature has been claimed able to be used as parameter for face recognition. This study aims to analyze the performance of geometric features, color features, and both of them on the human face using Gaussian Naïve Bayes (GNB) and the other Machine Learning method. This study using various geometric features: the distance between the eyes, nose, mouth by using Euclidean distance, and classified using GNB, K-Nearest Neighbour (KNN), and Support Vector Machine (SVM). The result compared with color feature: normalized RGB values, mean of normalized RGB, and RGB Variant as color features. The feature values obtained are assembled and processed using GNB and the other ML method to classified and recognized the faces. The dataset obtained from Aberdeen faces the dataset, which has 687 color faces from Ian Crow at Aberdeen. Between 1 and 18 images of 90 individuals. Some variations in lighting, varied viewpoints, and the resolution have varied between 336x480 to 624x544. The experimental results show that the system successfully recognized the face based on the determined algorithm and based on three models, SVM reached nearly 74.83%, GNB reached nearly 74.67%, and KNN with K = 5 reached nearly 72.17%.

Keywords: Face recognition, Geometric feature, Color feature, Machine learning techniques.

1. Introduction

Face recognition in images, in real-time video, or offline video are three of the most studied topics in computer vision [1]. Facial recognition is a way of recognizing a human face through technology. As the information on human identity, human faces are unique, and the face recognition system uses faceprint values as a feature from a photograph or video. It compares the information from trained data to find a match. Face recognition has used for various purposes, such as identification of people, supervision, robotics, and intelligent environments, thus high accuracy for detecting the face and recognize the face is the most crucial factor for face recognition [2] and [3]. Facial recognition can help verify personal identity, but it still raises many privacy issues. This study aims to find the pros and cons by using geometric feature and color feature.

The color feature has a drawback if a dataset has many labels with skin colors that tend to be the same as they come from the same tribe, then the classification results obtained will be reduced, so we need another feature model that is based on geometry, which extracts the value of facial features such as the distance between the eyes, nose and mouth. The proposed technique was a combination between well-known existing technique, named geometric facial features and color features like normalized Red (R), normalized Green (G), and normalized Blue (B), mean of RGB, and RGB variant in face area in order to compared the accuracy between color feature, geometry feature, and both of these features. GNB, KNN, and SVM were used as a method for classifying the data. The Aberdeen face database was selected for experiments to evaluate the face recognition algorithm proposed because the database has color faces and some variations in lightning and resolution. The face recognition system in this study

has been developed based on java netbeans and python for geometric features and color features. This system uses images as input, while the output is an image classification result.

This research aims to find the accuracy of the system. K Fold Cross validation with $k = 10$ was used to obtain the average of accuracy. The images were pre-processed with scaling the images dataset, Viola-Jones (VJ) was used to detect the face area and facial part area detection like right eye region, left eye region, nose region, and mouth region. Then the distances were calculated between midpoint of facial part area using Eculidean Distance (ED) to obtained the geometric feature. For color feature from face area detected using VJ, we extract the RGB color maps and preprocess it with normalized RGB for make it more robust from illuminance factor [4] and [5].

This paper was structured as follows, chapter 2 defines the review of existing techniques, which are used to detect the face, face parts, geometric features, color features, and classification methods. The proposed methodology is represented in chapter 3. The experimental result and from the machine Learning method is shown in chapter 4. Finally, the conclusion of the paper was represented in chapter 5.

2. Literature review

2.1 Related works

Face detection research has been done by [2] using template matching method and have successfully detected the face with good accuracy and research from [6] with the method of Hidden Markov Models (HMM) and dipped an accuracy of 95.9%. In face recognition field, [7] using Local Binary Pattern (LBP) and Principal Component Analysis (PCA) with 94% accuracy in low resolution and [8] has a survey of 2D face recognition techniques based on Global approaches, local approaches, and hybrid approaches, but didn't have a comparison between geometric distance feature between face parts and color feature in face region. Based on the problems and suggestions from the previous studies, the models for detecting, feature extraction, and preprocessing the dataset needs to be improved to obtain optimal results under different conditions. Currently, we can find a method for performing machine perceptions that are capable of solving the above problems. For face detection besides Neural Network (NN) based, there was an old but still robust and became a trend in non-deep learning method, called Viola-Jones (VJ) [9–11]. Preprocessing data by rescaling and using

normalized RGB space color [12] and [13] claimed can make the feature dataset become better. One of the methods for optimization is NBC which has successfully demonstrated its performance to perform the classification process in several fields, including pattern recognition, text classification, data mining, and bioinformatics [14-17]. The GNB was specialized for many parameters, many classes, and for numeric type features. The research with the GNB method with pre-processing techniques like resize, rescaling and normalized RGB thought to improve the accuracy of facial recognition with geometric and color features, and compared GNB results with another Machine Learning method, like KNN and SVM

2.2 Face detection using viola-jones

The trend of object detection, especially face detection, continues to develop, however, face detection using VJ by Paul Viola and Michael Jones since 2001 has remained a reliable detection method that can compete with deep learning-based object detection methods using neural networks or other convolutional neural networks [9-11]. Viola-Jones is designed to detect faces frontally. This algorithm detects faces in grayscale images and then finds their location in colored images. There are two stages in this algorithm, the first is training: training the classifier and using Adaptive Boosting (AdaBoost). The second is detection: detecting the haar-like feature and creating an integral image. The pre-trained model framework that can be obtained from many opensource fora named haarcascade_frontalface_default.xml was used to detect the face and facial parts like right eye, nose, and mouth. The result of the face and facial parts detection using VJ is shown in Fig. 1



Figure. 1 Detect the face and face parts using VJ

2.3 Geometric feature

Trend of face features extraction usually used biometric characteristic to recognize a person [8]. The geometry feature-based used to calculating the distance of facial parts features, such right eyes, left eyes, nose, and mouth, and their geometric relationships[18]. These feature points used as input data for the geometry-based application model. To get the distance value, midpoint of bounding box area of face part features must be calculated first, and can be observed in Eq. (1) [19].

$$(x, y) = (x_1 + \frac{width}{2}, y_1 + \frac{height}{2}) \quad (1)$$

(x, y) is the midpoint of the region, x_1 is the first coordinate of face-part region in the x-axis, y_1 is the first coordinate of face-part region in the y-axis. After the midpoint in facial parts region was obtained, the distance between the midpoints of the right eye, left eye, nose and mouth was obtained using Euclidean Distance (ED) so the six parameters including the distance of right eye to left eye, the distance of right eye to nose, the distance of left eye to nose, the distance of right eye to mouth, the distance of left eye to mouth, and the distance of nose to the mouth can be obtained, the result is shown in Fig. 2 and the ED equation can be observed in Eq. (2). x_1 is first feature midpoint coordinate in x-axis and x_2 is second feature midpoint coordinate in x-axis. y_1 is first feature midpoint coordinate in y-axis and y_2 is second feature midpoint coordinate in y-axis.

$$ED = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2)$$

2.4 Color feature

Red, Green, and Blue (RGB) is the basic color that exists on a computer system. The data in an image separated into components for each color. The color of each pixel determined by the combination of

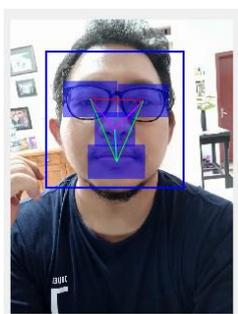


Figure. 2 Mid-point face-part detection and calculate the distance between mid-point facial features using ED

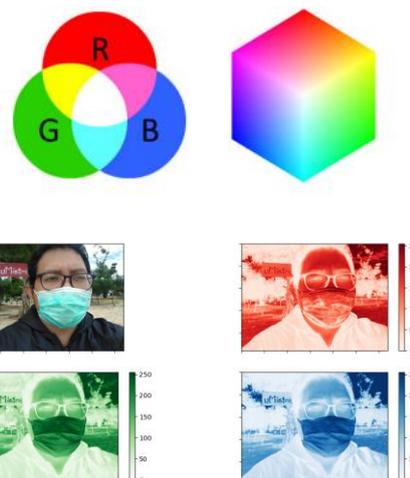


Figure. 3 RGB space color

the intensity of each color component and mapped in the coordinate form as shown in Fig. 3.

The color features previously presented in the journal by [20] resulted in differences in accuracy between the color spaces ranging from 69.44% using the XYZ color space, 71.50% using RGB, and $L^*a^*b^*$ with the highest accuracy of 79.77%, so the RGB base color was still feasible and can be used in facial recognition. We used VJ to detect the face region, like the first part in Fig. 1, then the face region was cropped and extracted the color feature like R, G, and B. We pre-processed the RGB value to normalized RGB to make it immune to luminance factor and can be observed in Eq. (3) from [2, 20, 21].

$$\begin{aligned} \text{Normalized } R &= \frac{R}{R+G+B} \\ \text{Normalized } G &= \frac{G}{R+G+B} \\ \text{Normalized } B &= \frac{B}{R+G+B} \end{aligned} \quad (3)$$

The average of RGB and the variant of RGB were added as parameter and a total of five parameters for color features were used.

2.5 Gaussian naïve bayes classifier (GNB)

The Naïve Bayes algorithm is a classification method using the probability and statistical methods proposed by the English scientist Thomas Bayes. This algorithm predicts future opportunities based on experience and is known as Bayes Theorem. The main feature of the Naïve Bayes Classifier (NBC) is a very strong or naive assumption of the independence for each parameter, condition, or event.

Naïve Bayes Classifier works very well compared to other classifier models. This sentence is

proved in the journal by [4], which says that Naïve Bayes Classifier has better accuracy than other classifier models.

The advantage of using this method is it requires only a small amount of training data to determine the estimated parameters required in the classification process, since it is assumed to be an independent variable, only the variant of a variable in a class is required to determine the classification. The common Naïve Bayes Formula expressed in Eq. (4), where $P(C|X)$ is posterior, $P(C)P(X)$ is likelihood, $P(C)$ is prior, and $P(X)$ is evidence.

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \tag{4}$$

In this study the data input varies in number with continuous data of parameter and multi class, so we use GNB instead and the formula expressed in Eq. (5) from [14–17] and [22].

$$P = (X_i = x_i | Y = y_i) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(x_i - u_{ij})^2}{2\sigma_{ij}^2}} \tag{5}$$

Where P is the probability, X_i is the I-parameter, x_i is the value of the 1st parameter, Y is the label we are looking for, ij is the Y sub-label you we are looking for, u is the mean, expressed the average of all the parameters and σ is the standard deviation of all parameters.

2.6 K-nearest neighbour (KNN)

KNN is a classifier where each pixel is classified in the same class as the training data with the closest intensity [12]. There is a possibility of an NN classifier produced an error decision if the obtained single neighbour is an outlier of some other class. To avoid this and improve the robustness of the approach the KNN classifier works with K patterns. The KNN classifier is considered a non-parametric classifier since it makes no underlying assumption about the statistical structure of the data. K-Nearest Neighbor (KNN) is a method that groups data based on the proximity of data to other data. One of the calculations of the distance between two objects that can be used is the euclidean distance. For example, object A with observation value $x = (x_1, x_2, \dots, x_p)$ and object B with observation value $y = (y_1, y_2, \dots, y_3)$, the euclidean distance between objects A and B which shown in Eq. (6):

$$d_{AB} = d(x, y) = \sqrt{\sum_{i=1}^p (x_i - y_i)^2} \tag{6}$$

where d_{AB} is the euclid distance between objects A and B and p is the number of variables that represent objects A and B .

The selection of the correct k value is very important in the KNN method. How to determine the best k value can be done by trial and error, namely by experimenting with several values of in order to obtain a model with a value of which produces the highest accuracy [20], in this experiment the k value used was 5.

2.7 Support vector machine (SVM)

In machine learning, Support Vector Machine (SVM) is a supervised learning model in analyzing data and pattern recognition which is used for classification and regression analysis. SVM is a method that works on the principle of Structural Risk Minimization (SRM) with the aim of finding the best hyperplane that separates two classes in the input space, as stated by Cortes and Vapnik, that: "Support Vector Machine (SVM) originally separates the binary classes ($k = 2$) with a maximized margin criterion."

Each training data is represented by (x_i, y_i) , where $i = 1, 2, 3, \dots, N$, so N is the amount of data. The available data is denoted as: $\rightarrow \in E^d$, where $x_i = \{x_{i1}, x_{i2}, x_{i3}, \dots, x_{iq}\}^T$ is the attribute (feature) for the i th training data. Meanwhile, each label / class is denoted as: where $i = 1, 2, 3, \dots, N$. It is assumed that both classes -1 and $+1$ can be completely separated by a hyperplane with d dimension, which is defined by Eq. (7).

$$w \cdot x_i + b = 0 \tag{7}$$

Where: w and b are model parameters for which you want to find the value. $w \cdot x_i$ is the inner product between w and x_i . Data / pattern x_i that is included in class -1 can be formulated as a pattern that satisfies the inequality in Eq. (8) and Eq. (9).

$$w \cdot x_i + b \leq -1 \tag{8}$$

$$w \cdot x_i + b \geq +1 \tag{9}$$

The SVM classification in Eq. (8) and Eq. (9) can be combined with a notation $y_i(w \cdot x_i + b) \geq 1, i = 1, 2 \dots N$. The largest / optimal margin can be found by maximizing the value of the distance between the hyperplane and its nearest point / data. Furthermore, this problem is formulated into a quadratic programming (QP) problem and can be seen in Eq. (10).

$$\min_w \tau(w) = \frac{1}{2} \|\vec{w}\|^2 \quad (10)$$

With requirements:

$$\begin{aligned} y_i(w \cdot x_i + b) - 1 &\geq 0 \\ y_i(w \cdot x_i + b) &\geq 1, i = 1, 2, \dots, N \end{aligned} \quad (11)$$

This optimization can be completed with the Lagrange Multiplier in Eq. (12):

$$Lp = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i y_i(w \cdot x_i + b) - 1 \quad (12)$$

The explanation above is based on the assumption that the two classes can be separated perfectly by hyperplane. However, generally the two classes in the input space cannot be separated completely. This causes the optimization process cannot be completed because there are no w and b that fulfill the Inequality Eq. (11). Thus, the inequality is modified by introducing the soft margin technique. In soft margin, the inequality is modified by including the slack variable $\xi_i (\xi_i \geq 0)$, Likewise with the equation Eq. (10).

$$w \in R^d, b \in R, \xi \in R = \frac{1}{2} \|w\|^2 + c \sum_{i=1}^N \xi_i \quad (13)$$

With requirement:

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i \quad (14)$$

Note: $w \in R^d$ is a weight vector, $C \in R$ used to control the trade off between margin and slack classification errors. The parameter C is useful for controlling the trade-off between margin and classification error. A large C value is a greater penalty for the classification error.

Like the Perceptron, SVM is actually a linear hyperplane which only works on data that can be linearly separated. For data whose class distribution is not linear, the Kernel approach is used in the initial data feature of the dataset. C in Eq. (13) is a regularization constant and a function of mapping training data to a suitable feature space. So it is possible to solve non-linear problems. Kernel functions that are usually used in SVM, namely

$$K(x_i, x_j) = x_i \cdot x_j \quad (\text{Linear SVM}) \quad (15)$$

$$K(x_i, x_j) = \exp\left(-\frac{|x_i \cdot x_j|^2}{2\alpha^2}\right) \quad (\text{RBF}) \quad (16)$$

The approach / strategy used in this experiment is One-Versus-One (1V1) or paired decomposition.

This approach evaluates all classifiers that may be paired and thus induces binary classification. Each classifier is applied to the test data which will provide one value for the winning class [13, 18, 19, 23, 24].

3. Proposed methodology

3.1 Dataset

Data collection for the dataset was done in accordance with the sources and types of data needed. In this study, data collection was Obtained from Aberdeen 2D face sets, which have color faces type dataset from Ian Crow at Aberdeen. The dataset has 1 to 18 images of 90 individuals. Has some variations in luminance or lighting, and variation of viewpoints. Resolution varied: 336x480 to 624x544. The part of dataset is shown in Fig. 4. Some face parts images maybe can't be detected using VJ since it can't detect the non-frontal face, so the one who can't be detected using VJ were eliminated.

3.2 Data acquisition

Data acquisition was the initial stage for obtained digital image parameters. The goal is to determine the data needed and choose digital image recording methods. In this stage, the dataset was extracted by geometric features and color features using java NetBeans and classified, the feature data extracted and saved into CSV format, then classified using GNB, KNN with $K = 5$, and SVM with RBF/Gaussian Kernel. The Result processed further with K-Fold Cross Validation with $K = 10$ to get mean accuracy and each ML Method results were compared.

3.3 Data preprocessed

Data pre-processed aims to simplify the process of identifying images. This pre-processed stage consists of resized and scalled into same size from the original image. Then for color feature we used normalized R, G and B from Eq. (3) to make the result more robust from luminance factor and added the average of RGB and variance of RGB as color features. The process of rescaled and feature extraction shown in Fig. 5 below.



Figure. 4 Aberdeen 2D face dataset

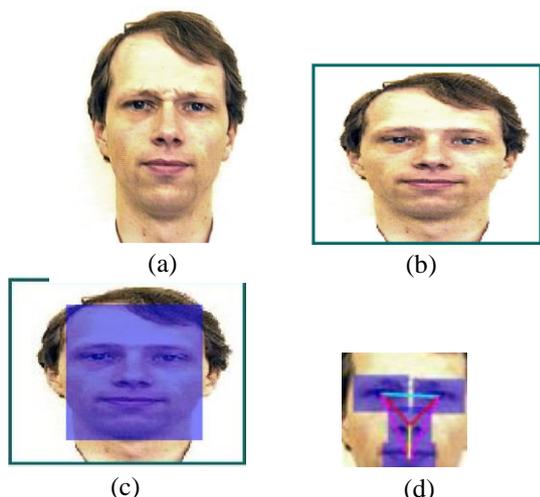


Figure. 5 Proposed technique for pre-processing images in dataset: (a) original image 384 X 512, (b) rescaled image 153 X 121, (c) face detected, and (d) geometric and color feature extraction

3.4 User Interface and System Design

The user interface in this study can be observed in Fig. 6. Data obtained from sources are processed using the proposed technique, and the system design for proposed method can be observed in Fig. 7. The data obtained normalized and extracted with proposed method. The model contained geometric, color, and geometric + color feature and classified with GNB, KNN, k = 5 and SVM one vs one.

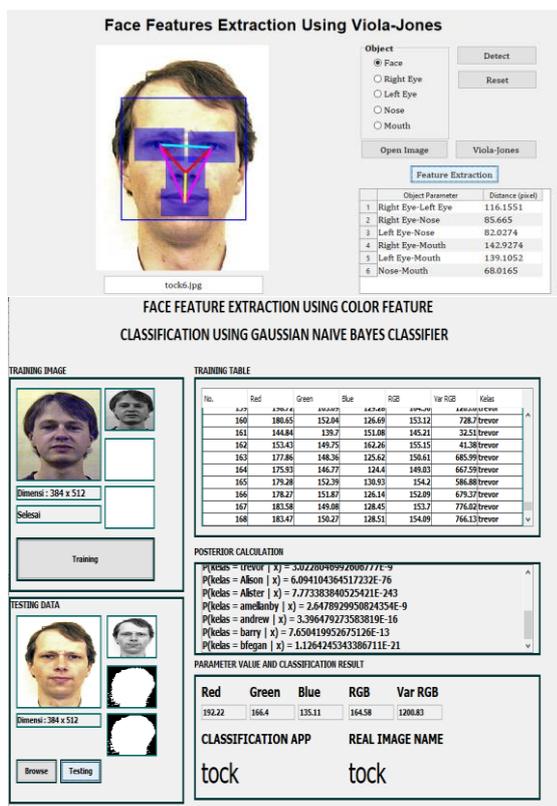


Figure. 6 System UI for feature extraction

3.5 Performance measurement method

The performance measure in this research using accuracy with k-fold cross validation, with k = 10. Cross Validation (CV) is a statistical method that can be used to evaluate the performance of a model or algorithm where the dataset is separated into two subsets, namely training data and validation or testing data. The model or algorithm is trained by the training subset and tested by the testing subset. Selection of CV type can be based on the size of the dataset. CV K-fold is used because it can maintain the estimation of accuracy. The K-Fold CV scheme that is generally often used is the 10-fold and was depicted in Fig. 8. The white block is a training-datasets and the red

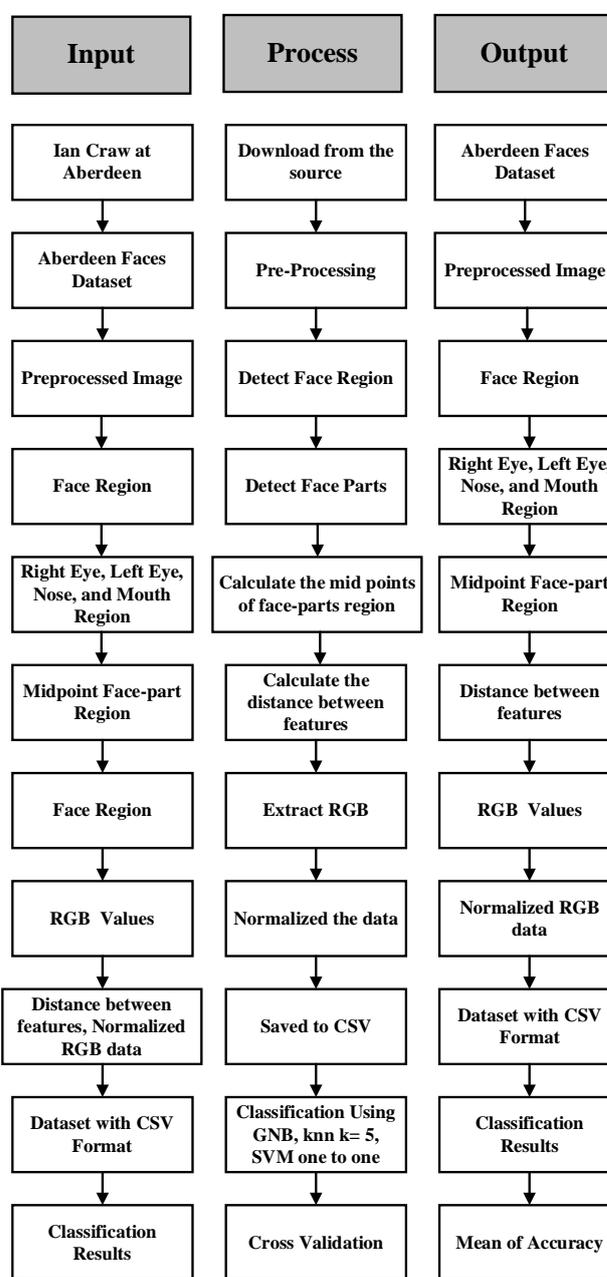


Figure. 7 System design

1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10

Figure. 8 10-fold CV

block is testing-datasets. The equation of accuracy can be observed in the Eq. (9). *TP* is True Positive, *TN* is True Negative, *FP* is False Positive, and *FN* is False Negative.

$$Accuracy = \frac{TN+TP}{TP+TN+FN+FP} \times 100 \quad (9)$$

4. Results and discussion

The result of testing graph can be seen in Fig. 9 for geometric feature and Fig. 11 for color feature. Based on these figures, scenarios are implemented in the dataset so that the results obtained from the accuracy of testing the scenarios in each fold. Chapter detailed the research relust and discussion of the proposed system. The proposed system was done using java NetBeans for feature extraction and Python for classification. The performance of geometric feature was compared with color feature and both geometric + color feature in order to compared the accuracy result of proposed system. With accuracy and 10-fold CV.

4.1 Performance analysis with geometric feature

The results analysis based on geometric features indicated that the accuracy value in each fold was kind of stable, and can be observed in Fig. 9. The average of 10-fold cross validation for geometric feature shown that SVM with RBF/Gaussian Kernel

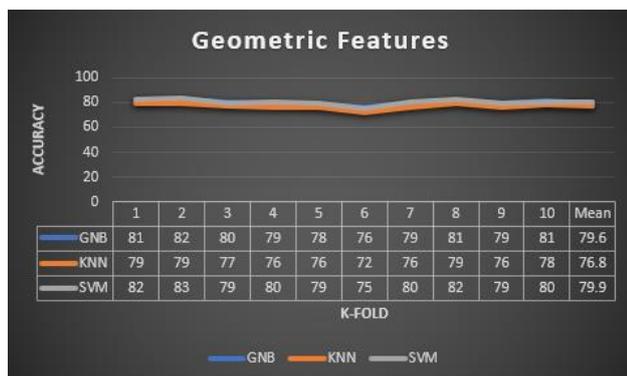


Figure. 9 Accuracy results by geometric features



Figure. 10 Unused images in dataset which can't be detected using VJ

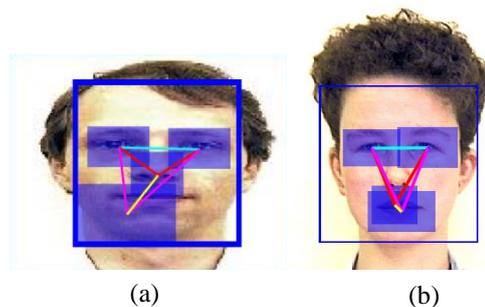


Figure. 11 The obstacle of face part detection using VJ: (a) miss detected in rescaled image and (b) miss detected in original image

has the highest accuracy with mean accuracy of 79,9% than KNN k = 5 with mean accuracy of 76,8%, and GNB with mean accuracy of 79,6% for face recognition.

The obstacle of face recognition by using geometric feature including if the face region and face-part region was not detected properly, the parameter would differ so much and will false classified. VJ for face detection specialized for frontal face only, so the first obstacle by using this feature is non-frontal image in dataset can't be used for geometric feature, this obstacle can be observed in Fig. 10.

The second obstacle for face recognition using geometric feature with VJ was the face part region didn't correctly detect the right region, like eyebrows was misidentified as eyes, mouth misidentified as nose, or can't be detected at all, so the parameter value will much different than the other same class dataset and reduced the accuracy value. This second obstacle can be observed in Fig. 11

Fig.11 (a) shown that and VJ can't detect face part well from image with low pixel, in this research the experiment scaled the images and can't worked well if the image scaled below 336 x 480 pixel. VJ still can detect the face, but not for face parts if the image size is too small. Fig. 11 (b) shown that VJ somehow miss identified the face part although original image has normal size between 336x480 to 624x544. Based on this experiment, geometric feature has important

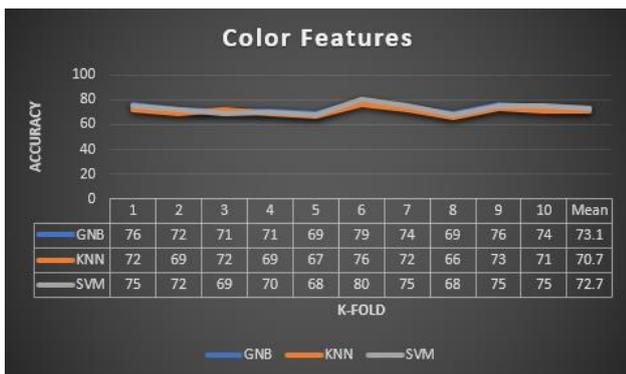


Figure. 12 Accuracy results by color features

factor to make it work well, like the image must have same size before the feature extracted and all of its feature must be extracted correctly to have optimum results.

4.2 Performance analysis with color features

The results analysis based on color features indicated that the accuracy value in each fold was stable, and can be observed in Fig. 12. The average of 10-fold cross validation for color feature was shown that GNB has the highest accuracy with mean accuracy of 73.1% than KNN k = 5 with mean accuracy of 70.7%, and SVM one to one with mean accuracy of 72.7% for face recognition.

The obstacle of face recognition by using color feature including the noise in background and the minimum variation in skin color. While we detecting the face using VJ and cropped the face region, a little bit of background color and hair color, also extracted and made the parameter value would not stable and often make false classified result and it can be normalized by scaling the image size. The example of noise using scaling and without scaling can be observed in Fig. 13 and Fig. 14



Figure. 13 Face region in original image



Figure. 14 Face region in rescaled image



(a) (b)
Figure. 15 Luminance factor in image

The usage of normalization the images dataset by scaled can reduced background and hair color, and make the data obtained have better quality than original image. The second obstacle was the difference of illuminance. Although normalized RGB claimed immune to luminance factor, the experiment result shown that it can reduce the illuminance factor, but still have an impact for classification by increase the false classification results. The detail information can be observed in Fig. 15.

Fig. 15 (a) have color value R = 136,99, G = 129,98, B = 133,44, RGB = 133,47, and Var RGB = 729,63 while Fig. 15 (b) have color value R = 161.14, G = 132.88, B = 104,43, RGB = 132,82, and Var RGB = 1120,66. The difference illumination within same class made the color feature have some kinds of weakness in illuminance factor, and increased the false classification results.

4.3 Performance analysis with geometric + color features

The results analysis based on geometric + color features can be observed in Fig.16. The average of 10-fold cross validation for color feature was shown that SVM has the highest accuracy with mean accuracy of 71.9% than KNN k = 5 with mean accuracy of 63%, and GNB with mean accuracy of 71.3% for face recognition.

The experiment results showed the combination between geometric and color features has lower accuracy than experiment with geometric feature

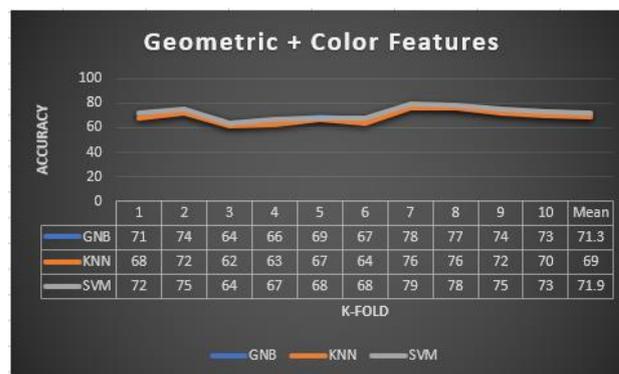


Figure. 16 Accuracy results by geometry + color features

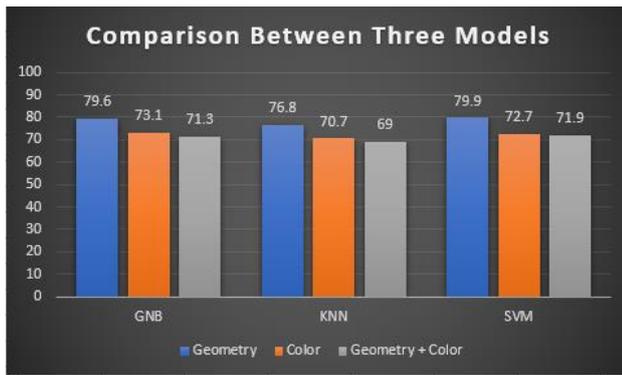


Figure. 17 Comparison results between three models

only, because the part of data in geometric features which the face part detected correctly combined with the data from color features which has significant difference in illuminance factor and have a little background color extracted slightly reduced the accuracy.

4.4 Comparison performance analysis from the experiment

The comparison result between three models can be observed in Fig. 17 below.

From the data above, face recognition from Aberdeen 2D face dataset with normalized and chosen image which the features can be detected perfectly with VJ shown that GNB has mean accuracy by 74.67 %, KNN with K=5 has mean accuracy by 72.1, and SVM has mean accuracy by 74.83.

The comparative study of proposed and existing work from [26]. [26] developed face recognition with enhanced firefly optimization algorithm and deep belief network from YALE, ORL, and FASSEG dataset which have similarity with Aberdeen dataset and have similar results for face recognition by using SVM and KNN, shown in Table 1. The SVM method with KNN using enhanced firefly optimization has an accuracy value that is not much different from the proposed method, the KNN method is used using KNN with a value of $k = 5$, if k with a smaller value is applied, the accuracy will be higher.

The accuracy result in experiment seem good, but in real life, the condition which is dynamic is not same as the dataset which is static. not all good condition can be found. From the face detection problem, the non-frontal image, the distance between

Table 1 Performance evaluation of proposed work with similar performance measures

Classifier	ORL Database	Yale Database	FASSEG Database	Aberdeen Database
SVM	70.98 %	67.20 %	68.18 %	74.83 %
KNN	79 %	82.30 %	84.98 %	72.1 %

face and camera, variation in expression, using accessories in face, heavy makeup, and occlusion can make the face cant be detected or can be detected, but the feature inside the face will varied heavily, so the proposed model can be used for static dataset only, like driver license dataset or identity card, with optimum result if the dataset frontal face, same lightning, same distance, same pose, same expression, no heavy face accessories, no heavy makeup, and no occlusion in the process for making dataset. For dynamic environment, the proposed method needs to be enhanced further with more parameters.

5. Conclusion

Face Recognition from Aberdeen dataset has successfully implemented by geometric feature, color features, and geometric + color feature on the faces of Aberdeen datasets and classify them according to their labels. The advantage of the color feature is that in the dataset, there are many people with different races and static conditions such as the Aberdeen dataset, and a geometric feature with six parameters that contains the distance between the eyes, nose, and mouth can improve accuracy in a dataset that has little racial variation. When these two features are combined, the results will be even more accurate even its lower than pure geometric feature. The features used in this dataset can work in static environments such as driving license data or population data but need to be improved to be able to recognize faces with dynamic conditions The mean accuracy between three models generate GNB with an average accuracy of 74.67 %, KNN K=5 w with an average accuracy of 72.1%, and SVM one to one with an average accuracy of 74.83.

In real life, not all good condition can be found, here are many varies, so it far from good if face recognition implemented by proposed method and the dataset has many variations. The good condition must be set to gained an optimum accuracy.

Conflicts of Interest

The author(s) declare(s) that there is no conflict of interest regarding the publication of this article.

Author Contributions

“Conceptualization, associate professor Cahya Rahmad and associate professor Rosa Andre; methodology, associate professor Cahya Rahmad; software, Dimas Rossiawan; validation, associate professor Ekojono, associate professor Rosa Andre, and associate professor Cahya Rahmad; formal analysis, associate professor Cahya Rahmad,

professor Kohei Arai; investigation, associate professor Cahya Rahmad; resources, Dimas Rossiawan; writing—original draft preparation, Dimas Rossiawan; writing—review and editing, Dimas Rossiawan; visualization, associate professor Cahya Rahmad; supervision, professor Kohei Arai; project administration, associate professor Cahya rahmad, associate professor Rosa Andre; funding acquisition, associate professor Cahya Rahmad from Indonesian Government : Ristekdikti”.

References

- [1] C. Gonzalez Viejo, S. Fuentes, K. Howell, D. D. Torrico, and F. R. Dunshea, “Integration of non-invasive biometrics with sensory analysis techniques to assess acceptability of beer by consumers”, *Physiol. Behav.*, Vol. 200, No. 2017, pp. 139–147, 2019.
- [2] R. Wijanarko *et al.*, “Faces detection method based on skin color modeling”, In: *Proc. of - 2017 Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discov. CyberC 2017*, Vol. 4, No. 1, pp. 40–49, 2017.
- [3] M. C. Fachrudik, C. Rahmad, and B. Harijanto, “Rancang Bangun Sistem Pengenalan Wajah Untuk Identifikasi Mahasiswa Yang Melakukan Pelanggaran Melalui Bukti Foto Menggunakan Metode Eigenface”, *J. Inform. Polinema*, Vol. 3, No. 1, p. 1, 2018.
- [4] Z. Xu, Y. Jiang, Y. Wang, Y. Zhou, W. Li, and Q. Liao, “Local polynomial contrast binary patterns for face recognition”, *Neurocomputing*, Vol. 355, pp. 1–12, 2019.
- [5] C. J. Du and D. W. Sun, “Comparison of three methods for classification of pizza topping using different colour space transformations”, *J. Food Eng.*, Vol. 68, No. 3, pp. 277–287, 2005.
- [6] N. W. Pratiwi, F. Fauziah, S. Andryana, and A. Gunaryati, “Deteksi Wajah Menggunakan Hidden Markov Model (HMM) Berbasis Matlab,” *STRING (Satuan Tulisan Ris. dan Inov. Teknol.*, Vol. 3, No. 1, p. 44, 2018.
- [7] K. Hidayati, A. B. Utomo, and H. Wibawanto, “Pengenalan Wajah Menggunakan Metode Local Binary Pattern (LBP) dan Principal Component Analysis (PCA) untuk Citra Berkualitas Buruk”, *e-Proceeding Eng.*, Vol. 5, No. 3, pp. 7928–7938, 2018.
- [8] M. Chihaoui, A. Elkefi, W. Bellil, and C. Ben Amar, “A survey of 2D face recognition techniques”, *Computers*, Vol. 5, No. 4, pp. 1–28, 2016.
- [9] D. I. S. Saputra, R. A. Pamungkas, K. A. N. Ramadhan, and W. S. Anjar, “Pelacakan Dan Deteksi Wajah Menggunakan Video Langsung Pada Webcam”, *Telematika*, Vol. 10, No. 1, pp. 50–59, 2017.
- [10] Jeetu, “F5131086617”, No. 6, pp. 101–106, 2017.
- [11] K. Vikram and S. Padmavathi, “Facial parts detection using Viola Jones algorithm”, In: *Proc. of 2017 4th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2017*, pp. 2015–2018, 2017.
- [12] C. Rahmad, I. F. Rahmah, R. A. Asmara, and S. Adhisuwignjo, “Indonesian traffic sign detection and recognition using color and texture feature extraction and SVM classifier”, In: *Proc. of 2018 Int. Conf. Inf. Commun. Technol. ICOIACT 2018*, Vol. 2018-Janua, No. c, pp. 50–55, 2018.
- [13] R. A. Asmara, F. Rahutomo, Q. Hasanah, and C. Rahmad, “Chicken meat freshness identification using the histogram color feature”, In: *Proc. of - 2017 Int. Conf. Sustain. Inf. Eng. Technol. SIET 2017*, Vol. 2018-Janua, No. c, pp. 57–61, 2018.
- [14] B. M. Gayathri and C. P. Sumathi, “An Automated Technique using Gaussian Naive Bayes Classifier to Classify Breast Cancer”, *Int. J. Comput. Appl.*, Vol. 148, No. 6, pp. 16–21, 2016.
- [15] U. Widodo Wijayanto and R. Sarno, “An Experimental Study of Supervised Sentiment Analysis Using Gaussian Naïve Bayes”, In: *Proc. of - 2018 Int. Semin. Appl. Technol. Inf. Commun. Creat. Technol. Hum. Life, iSemantic 2018*, pp. 476–481, 2018.
- [16] A. A. Rafique, A. Jalal, and A. Ahmed, “Scene Understanding and Recognition: Statistical Segmented Model using Geometrical Features and Gaussian Naïve Bayes”, In: *Proc. of 2019 Int. Conf. Appl. Eng. Math. ICAEM 2019 - Proc.*, No. June, pp. 225–230, 2019.
- [17] N. Sehe, M. S. Lew, I. Cohen, A. Garg, and T. S. Huang, “Emotion recognition using a cauchy naive bayes classifier”, In: *Proc. of - Int. Conf. Pattern Recognit.*, Vol. 16, No. 1, pp. 17–20, 2002.
- [18] A. Agrawal, “Comparative Study of Lip Extraction Feature with Eye Feature Extraction Algorithm for Face Recognition”, *Int. J. Comput. Appl. Technol. Res.*, Vol. 5, No. 6, pp. 330–337, 2016.
- [19] N. Sabri *et al.*, “A comparison of face detection classifier using facial geometry distance measure”, In: *Proc. of 2018 9th IEEE Control Syst. Grad. Res. Colloquium, ICSGRC 2018 - Proceeding*, No. August, pp. 116–120, 2019.
- [20] H.-I. Kim and Y. M. Ro, “COLLABORATIVE FACIAL COLOR FEATURE LEARNING OF

MULTIPLE COLOR SPACES FOR FACE RECOGNITION Hyung-II Kim and Yong Man Ro † Image and Video Systems Lab , School of Electrical Engineering , KAIST , Republic of Korea”, In: *Proc. of 2016 IEEE Int. Conf. Image Process.*, 2016.

- [21] G. Bao, S. Cai, L. Qi, Y. Xun, L. Zhang, and Q. Yang, “Multi-template matching algorithm for cucumber recognition in natural environment”, *Comput. Electron. Agric.*, Vol. 127, pp. 754–762, 2016.
- [22] J. Y. Choi, Y. M. Ro, and K. N. Plataniotis, “Boosting color feature selection for color face recognition”, *IEEE Trans. Image Process.*, Vol. 20, No. 5, pp. 1425–1434, 2011.
- [23] M. Ontivero-Ortega, A. Lage-Castellanos, G. Valente, R. Goebel, and M. Valdes-Sosa, “Fast Gaussian Naïve Bayes for searchlight classification analysis”, *Neuroimage*, Vol. 163, pp. 471–479, 2017.
- [24] Q. J. Wang and R. B. Zhang, “LPP-HOG: A new local image descriptor for fast human detection”, In: *Proc. of 2008 IEEE International Symposium on Knowledge Acquisition and Modeling Workshop Proceedings, KAM 2008*, pp. 640–643, 2008.
- [25] M. Ring and B. M. Eskofier, “An approximation of the Gaussian RBF kernel for efficient classification with SVMs”, *Pattern Recognit. Lett.*, Vol. 84, pp. 1339–1351, 2016.
- [26] P. Annamalai, “Automatic face recognition using enhanced firefly optimization algorithm and deep belief network”, *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 5, pp. 19–28, 2020.