2021 17th International Conference on Quality in Research (QIR): International Symposium on Electrical and Computer Engineering | 978-1-6654-9696-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/QIR54354.2021.9716171

Performance Evaluation of Machine Learning Classifiers for Face Recognition

Dodi Sudiana
Department of Electrical Engineering
Faculty of Engineering
Universitas Indonesia
Depok, Indonesia
dodi.sudiana@ui.ac.id
(corresponding author)

Mia Rizkinia
Department of Electrical Engineering
Faculty of Engineering
Universitas Indonesia
Depok, Indonesia
mia@ui.ac.id

Fahri Alamsyah
Department of Electrical Engineering
Faculty of Engineering
Universitas Indonesia
Depok, Indonesia
fahri.alamsyah@ui.ac.id

***Abstract*— The digital world, especially image processing, has been evolving due to the needs of society and the importance of digital-based system security. One of the rapidly progressing technologies is the face recognition system using artificial intelligence. It recognizes a person's face registered in the database for verification purposes. In this study, we evaluate the face recognition systems based on machine learning classifier algorithms and Principal Component Analysis (PCA) for feature extraction. Seven machine learning algorithms were considered, i.e., Support Vector Machine (SVM), Decision Tree, K-Nearest Neighbour (KNN), Logistic Regression, Naïve Bayes, Multi-Layer Perceptron (MLP), and Convolutional Neural network (CNN). In the CNN scenario, PCA was not used since it has its feature extraction capability. The first six classifiers were set to the most optimal settings. At the same time, CNN used the LeNet-5 architecture trained with a dropout rate of 0.25, 60 epochs, batch size of 20, Adam optimizer, and cross-categorical entropy for the loss function. The input image size was 64×64×1 obtained from the Olivetti faces database. CNN, SVM, and LR achieved the three highest accuracies, i.e., 98.75%, 98.75%, and 97.50%, respectively.**

Keywords*— *Face recognition, machine learning, Principal Component Analysis (PCA), Convolutional Neural network (CNN), performance evaluation.

I. Introduction

The face is the most memorable body part in real life and becomes an essential variable. It has been used to represent face images in Artificial Intelligence (AI) applications, especially face recognition. Facial imagery is used worldwide for citizenship, identity cards, law enforcement, user authentication, and more. This technology identifies each input image by comparing it with face images in the database.

AI-based face recognition developed when Facebook began implementing 120 million neurons which used 4 million images uploaded by Facebook users and succeeded with 97% accuracy. Besides Facebook, Google is also developing it under FaceNet as its architecture. Google created its use to recognize someone with face recognition with data from 260 million face sets with 86% accuracy [1].

The face recognition process consists of two main steps is feature extraction and classification [2]. The initial stage recognizes faces by calculating facial characteristics such as nose size, brow width, and forehead area. Face images can take much time to identify because they have substantial pixel sizes; hence, a dimensional reduction is required, usually accomplished by a feature selection or feature extraction process [3]. Feature extraction aims to convert face space into feature space. One of the feature extraction processes uses Principal Component Analysis (PCA) in face recognition. PCA is probably the most popular multivariate statistical technique used by almost all disciplines [4]. .

The next step, i.e., classification, aims to predict the class to which the input belongs, in other words, to identify the person. Some of the well-known classifier algorithms are Support Vector Machine (SVM), Decision Tree, K-Nearest Neighbor (KNN), Naïve Bayes, Logistic Regression, and Multi-Layer Perceptron (MLP). Several existing studies show that the accuracy of using specific algorithms is quite good with PCA [2].

Support Vector Machines (SVM) is one of the most valuable techniques in classification problems. Its application to computer vision has been proposed recently, such as face recognition [5]. SVM concept itself is to find the most optimal hyperplane that separates the data set into two classes linearly [6]. SVM has various kernels such as linear, quadratic, radial basis function, sigmoid, and others [7].

Decision Tree is a machine learning technique that builds a recursive representation of classification rules [6]. The decision tree starts from the roots and then finds the "best" attribute to test and partition the data further at each node. This process continues until it shows the classification decision [8].

K-Nearest Neighbor (KNN) is a machine learning method that looks for several k patterns (among all the training patterns in all classes) closest to the input pattern, then determines the decision class based on the highest number of practices among the k patterns [6]. If adequately trained, KNN has less bias than other algorithms [9].

Naïve Bayes is a machine learning method that learns from data and predicts classes in which each type has a probability [10]. Naïve Bayes is a classic method with a simple concept of probability but provides reasonably good performance for many modern cases with extensive data [6].

Logistic regression is a machine learning method that applies statistical techniques. There are two types, namely binary logistic regression analysis, and multinomial logistic regression. In binary type logistic regression analysis, the sigmoid function is usually applied. Multinomial logistic regression looks at the relationship between input and output data to create a model and generates weights for the classification results [11].

Multi-Layer Perceptron (MLP) is part of an Artificial Neural Network (ANN) whose structure consists of an input layer, a hidden layer, and an output layer [7]. Some critical parameters usually used in MLP are the number of hidden layers, the number of iterations for training and activation functions [12].

A Convolutional Neural network (CNN) is a machine learning method that develops MLP to process two-dimensional data and implement it on image data. CNN performs feature extraction and classification simultaneously

978-1-6654-9696-4/21/$31.00 ©2021 IEEE

on the layering process. Encoding an image into features in numbers representing the image is the initial feature extraction stage [13]. Some existing architectures include LeNet-5, AlexNet, and ZFNet, with each different configuration for each layer. CNN works by optimizing the current parameters using the back-propagation algorithm, and LeNet-5 combines feature extraction and image recognition [14].

This paper presents the performance comparison of these seven algorithms to evaluate which is the best to be implemented in the face recognition system. CNN requires a large dataset for the training process, so data augmentation is needed before the training phase. Data augmentation uses several techniques, including horizontal and vertical flipping of images, random cropping or images, color jitter or unexpected variations in contrast, and random combinations such as translation or reduction and rotation [6]. In addition, in the CNN scenario, dropout was set in the training process to prevent over-fitting, which can cause a decrease in model accuracy. The dropout layer value for face recognition is usually set to 0.25, 0.80, or 1 [15]. From several existing studies, the use of CNN has an accuracy of more than 90% [16].

The analysis will be carried out from face recognition results using a confusion matrix and classification report. These metrics contain information about the predicted results and actual data used to determine if an error occurs and see the value of precision, recall, and F1-Score. The rest of the paper consists of several parts: Section II describes the materials and methods used in this research. Section III describes the experiment results, and Section IV relates to conclusions and suggestions.

II. Materials And Methods

The face recognition system design determines whether the existing data is the same as the dataset on labels from 0-39btained from the Olivetti faces database [17]. The classifications to be compared are SVM, K-NN, Naïve Bayes, Logistic Regression, MLP, and Decision Tree. The most optimal parameters were previously determined by feature extraction using PCA to the input image. In addition, for the CNN implementation, the architecture used was LeNet-5, and the data augmentation was conducted before the model training.

The hardware specification used for the experiment was Intel (R) Core (TM) i5-8265U CPU @ 1.60GHz (8 CPUs), with 8GB RAM, 1TB HDD, and a 64-bit operating system. The software used is Google Collaboratory as a coding environment for the Python programming language.

A. Dataset

The dataset used is Olivetti Faces from Cambridge University Engineering Department, which is available in scikit learn with different dimensions [17]. The dataset consists of 400 grayscale images with a sample size of 64×64 pixels with ten image variations from 40 people. Pictures are taken at different times with variations in lighting, facial expressions (eyes open/closed, smiling/not smiling), and facial details (glasses/without glasses). In addition, all images use a dark homogeneous background with a perpendicular position. The label associated with each face image ranges from 0 to 39 and corresponds to the subject IDs.

B. Methods

Fig. 1 shows a flowchart of face recognition scenario design using SVM, Logistic Regression, MLP, Naïve Bayes, KNN, Decision Tree, and CNN as classifiers.

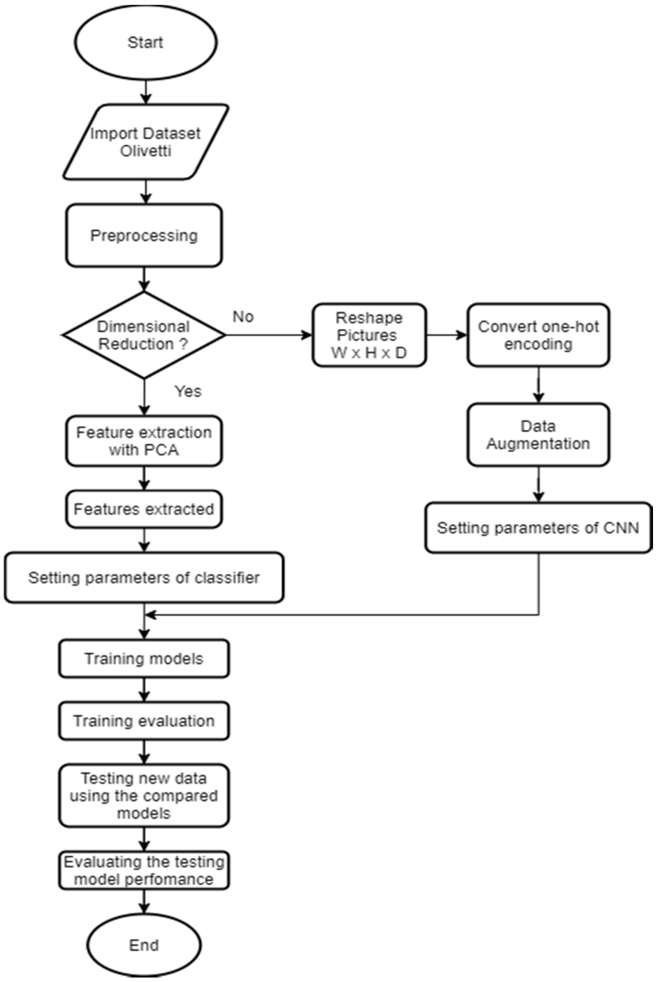

Fig. 1. Flowchart of the face recognition scenario design.

The first step is to import the Olivetti faces dataset from sklearn. The followed steps are:

1. Preprocessing: collecting ten variations of faces, separating training and testing data with a ratio of 80:20. In total, there were 320 training samples and 80 test samples.
2. The dimension reduction stage must go through feature extraction with PCA, and otherwise, it will enter the CNN process.
3. Feature extraction with PCA: finding the most optimal number of components with a data variance percentage of more than 90% while maintaining data authenticity.
4. Extracted features: the extracted features will be inputted to each classifier used.
5. Setting parameters of classifiers: each classifier was assigned to the above parameter setting. The classifiers are SVM, Logistic Regression, MLP, Naïve Bayes, KNN, Decision Tree, and CNN. All classifiers use the sci-kit-learn library, except CNN, which uses the Keras library.
6. Training models and evaluation: the next stage is to train each classifier to fit the training data. The models

generated from the training process were evaluated using confusion matrix, accuracy, precision, recall, and F1-Score.

7. Testing and evaluating the testing model performance: these steps aim to test and evaluate the generated models using different face identities with 40 classes for everyone in the dataset.

The initial stage of CNN is to reshape the image with dimensions W (Width)×H (Height)×D (Depth) to be 64×64×1. The data preprocessing step is one-hot encoding. One-Hot encoding is an encoding method in which data is a binary vector with an integer value of 40 classes on the face dataset. After that, data augmentation was performed with 5% zoom and 8 degrees rotation randomly to increase accuracy and build a CNN model with LeNet-5 architecture. The parameters used to perform the training model, i.e., Adam optimizer with a learning rate of 0.001 to minimize the categorical cross-entropy as loss function. The epoch value was set to 30, and the batch size was 20, which are determined as the optimum values after several experiments.

III. RESULT AND DISCUSSION

A. PCA Process Results

Before the PCA process, separating the training and testing data and the number of samples for each class is necessary. The initial process of PCA is to import the PCA library. The most crucial part of using PCA is estimating how many components are needed to describe the data. The next step is determining the optimal number of principal components by observing the relationship between the number of components and cumulative variance. As shown in Fig. 2, when the number of components is 70, the variance value has reached more than 90%, which covers most of the essential characteristics of the data. PCA for dimensionality reduction involves concentrating one or more minor principal components, resulting in a projection of the lower data dimensions that maintain the maximum data variance.

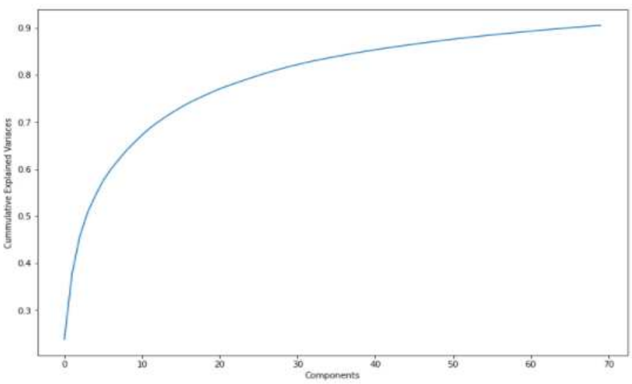

Fig. 2. Graph of variance with the number of components.

In other words, the stopping condition criterion used for PCA is 90% of the cumulative variance, where the component value reaches 70. If there is an additional component value, the increase is not too significant. The selection of the number of components represents the number of eigenface results. Eigenfaces help extract facial information related to the human perception of facial characteristics. The results are fascinating and varied, as shown in Fig. 3.

Fig. 3. Result of eigenfaces with the number of components 70.

The first few eigenfaces from the top left side were related to the lighting angle on the face, and then other vectors form certain features like eyes, nose, and lips. In addition, as the main component of the face distribution or identical to the covariance matrix of the face image collection. These eigenfaces are given as input to the classifier to detect individuals. The next stage is modeling with GridSearchCV, part of the scikit learn module that chooses the hyperparameter to provide the best performance model [18]. Each parameter in each classifier has been set to the optimal value.

B. Face Recognition Results

To get the result, we have different times to classifiers predict and train the data. Table I shows the training and prediction time of the experiments.

TABLE I. THE TRAINING AND PREDICTION TIME OF EACH METHOD

Algorithm	Time	
	Training (s)	Prediction (s)
PCA + SVM	3.984	0.005
PCA + Decision Tree	2.850	2.852
PCA + KNN	0.436	0.006
PCA + Logistic Regression	51.667	0.004
PCA + Naïve Bayes	**0.300**	0.002
PCA + MLP	80.301	**0.001**
CNN	265.003	0.517

According to Table I, Naïve Bayes becomes the fastest, and the CNN becomes the longest time during the training process. During the prediction process, the MLP becomes the fastest while PCA + Decision Tree becomes the longest. The prediction is more important than the training time in face recognition because face recognition systems must be fast and accurate at the same time in attendance, monitoring, and surveillance systems [19].

Testing is done with several classifiers to get accuracy and error values from image predictions and details. The SVM's accuracy was 98.75%, with one image showing false predictions. The detailed error is open eyes, smiling, slightly sideways position, not wearing glasses, and bright illumination. The accuracy achieved using the Decision Tree was 57.50%, with 34 images showing false predictions. Most of the mistakes made are open eyes, not smiling, slightly sideways position, not wearing glasses, and bright illumination. The accuracy achieved using KNN was 88.75%, with 9 images showing false predictions. Most of the mistakes made are eyes open, not smiling, normal position, not wearing glasses, and bright illumination. The accuracy achieved using Logistic Regression was 97.50%, with two images showing false predictions. The detailed error is open eyes, smiling, slightly sideways position, not wearing glasses, and bright illumination. The accuracy achieved using Naïve Bayes was 95.00%, with 4 images showing false predictions. Most of the mistakes made are open eyes, smiling, normal position, not wearing glasses, and bright illumination. The accuracy achieved using Multi-Layer Perceptron is 87.50%, with 10 images showing false predictions. Most of the mistakes made are open eyes, smiling, normal position, not wearing glasses, and bright illumination. CNN's accuracy was 98.75%, with one image showing false predictions. The detailed error is open eyes, smiling, slightly sideways position, not wearing glasses, and bright illumination.

C. *Analysis of Model Evaluation*

Analysis of model evaluation was conducted by observing the confusion matrix and the average of each precision, recall, and F1-Score. The details of the confusion matrix analysis are shown in Fig. 4.

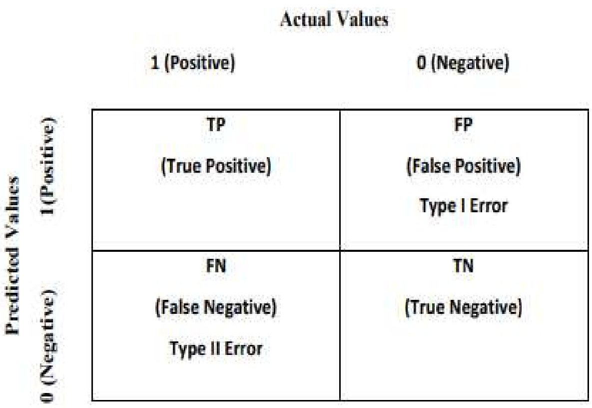

Fig. 4. Confusion Matrix

Precision and recall values are very influential in determining good models in their sensitivity to correctly predict positive cases among their positive predictions and all positive-labeled data, respectively. Formulas for accuracy, precision, recall, and F1-Score are formulated below:

$$\text{Accuracy} = \frac{TP}{The\ number\ of\ data} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{3}$$

$$\text{F1-Score} = \frac{2\ x\ Recall\ x\ Precision}{Recall+Precision} \tag{4}$$

In the face recognition task, FP predictions occur when the actual face data is not the ID label. Then the prediction model is generated several times for the face according to the ID label. Meanwhile, FN occurs when the input face images are not recognized and are saved in the labeled database. FN is more tolerable for face recognition tasks than FP since FN may be solved by taking the input image several times, while FP may become a security issue as it may allow a wrong person to be approved. The precision and accuracy parameters are presented in Table II.

TABLE II. THE AVERAGE AND ACCURACY OF EACH METHOD

Algorithm	Dataset: Olivetti (80:20)			
	Avg Precision	Avg Recall	Avg F1-Score	Accuracy
PCA + SVM	0.99	0.99	0.99	**0.988**
PCA + Decision Tree	0.60	0.57	0.56	0.575
PCA + KNN	0.92	0.89	0.88	0.888
PCA + Logistic Regression	0.99	0.97	0.97	0.975
PCA + Naïve Bayes	0.97	0.95	0.95	0.950
PCA + MLP	0.87	0.88	0.86	0.875
CNN	0.99	0.99	0.99	**0.988**

From Table II, it can be concluded that CNN and SVM obtain the highest accuracy with two different models, and then the lowest one is the Decision Tree. The average precision is included in each algorithm because it is almost close to 1. The error factor is incorrect predictions on ID 0, ID 2, ID 9, and ID 15. One can observe that their images have specific characteristics such as opened eyes, smiling or not smiling, no glasses, bright lighting. These can be the error factors that need to be avoided in future data collection. IDs that often cause low precision are ID 0 and ID 39. Meanwhile, IDs that often cause low recall are ID 0, ID 2, ID 4, ID 9, ID 10, ID 20, ID 26, and ID 35.

IV. CONCLUSIONS AND SUGGESTIONS

Face recognition system using PCA as feature extraction and machine learning algorithms as a classifier with the most optimal set of parameters resulted in varying accuracy. Meanwhile, CNN produced the best accuracy with LeNet-5 architecture, dropout rate value of 0.25, epochs value of 30, and batch size of 20 with loss function cross-categorical entropy, optimizer adam, and image input 64×64×1. The accuracy range obtained was 60%-99%, and 98.75% on average. In addition, the image prediction error for the dataset used, namely Olivetti, varies from 1 image to 34 images from 80 existing testing data. Variations of image errors consist of aspects of facial expression (smiling or not smiling), eyes (open or closed), face position (normal or slightly to the side), facial details (wearing glasses or not), and illumination. Suggestions for future research are to use more datasets and focus on one error, such as wearing glasses or not.

ACKNOWLEDGMENT

This research is funded by Universitas Indonesia Publikasi Terindeks Internasional (PUTI) Q2 Research Grant based on contract No. NKB-4326/UN2.RST/HKP.05.00/2020.

REFERENCES

[1] Yogalakshmi.S, "Review on digital image processing techniques for face recognition," International Conference on Communication and Signal Processing, vol. I, no. 1, pp. 1633-1637, 2020.

[2] S. Sharma, "Face recognition system using machine learning," International Conference on Communication and Electronics Systems (ICCES 2020), p. 1162, 2020.

[3] J. VanderPlas, Python data science handbook, United States Of America: O'Reilly Media, 2016.

[4] A. L. Ramadhani, P. Musa and E. P. Wibowo, "Human face recognition application using PCA and eigenface approach," in Second International Conference on Informatics and Computing (ICIC), Jayapura, Indonesia, 2017.

[5] C. Jain and R. K. Sahu, "A survey: face recognition techniques," International Journal for Research &, vol. V, no. 1, pp. 108 - 114, 2016.

[6] Suyanto, Basic and advanced machine learning, Bandung: Penerbit Informatika, 2018.

[7] R. C. Damale and B. V. Pathak, "Face Recognition Based Attendance System Using Machine Learning Algorithms," in Second International Conference on Intelligent Computing and Control Systems (ICICCS), India, 2018.

[8] S. S. Redla, D. B. Mallik and D. V. K. Mangalampalli, "Coefficient of variation based decision tree classifier for face recognition with invariant moments," in 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), India, 2020.

[9] X. Guo, "A KNN classifier for face recognition," in 3rd International Conference on Communications, Information System and Computer Engineering (CISCE 2021), China, 2021.

[10] E. B. Putranto, P. A. Situmorang and A. S. Girsang, "Face recognition using eigenface with Naïve Bayes," in 11th International Conference on Knowledge, Information and Creativity Support Systems (KICSS), Yogyakarta, 2016.

[11] S. Ongkittikul, J. Suwatcharakulthorn and K. Chutisowan, "Convolutional Multinomial Logistic Regression for Face Recognition," in 8th International Electrical Engineering Congress (iEECON), Chiang Mai, 2020.

[12] W. Ouarda, H. Trichili, A. M. Alimi and B. Solaiman, "MLP Neural network for face recognition based on Gabor features and dimensionality reduction techniques," in 2014 International Conference on Multimedia Computing and Systems (ICMCS), Marrakech, Maroco, 2014.

[13] A. Krizhevsky, "ImageNet classification with deep convolutional neural networks," Communications of the ACM, vol. 60, pp. 84 - 90, 2017.

[14] J. G. Guan Wang, "Facial expression recognition based on improved LeNet-5 CNN," in Chinese Control And Decision Conference (CCDC), China, 2019.

[15] D. W. Di Wang, "Face recognition system based on CNN," International Conference on Computer Information and Big Data Applications (CIBDA), pp. 1-2, 2020.

[16] J. Xiao and G. Weiwei, "Face recognition algorithm based on Prewitt and convolutional neural network," in IEEE, Xiamen, 2019.

[17] "scikit-learn," scikit-learn developers, 2007-2020. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_olivetti_faces.html. [Accessed 30 July 2021].

[18] M. Chandrakala and P. D. Devi, "Two-stage classifier for face recognition using HOG features," in Materials Today, 2021.

[19] S. Sourabh and S. Geeta, "A comparative study of classifiers used in facial embedding classification," in International Conference on Secure Cyber Computing and Communication (ICSCCC), India, 2018.