Indonesian Journal of Electrical Engineering and Computer Science
Vol. 23, No. 2, August 2021, pp. 811~820
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v23.i2.pp811-820

Cloud-based architecture for face identification with deep learning using convolutional neural network

Aditya Herlambang, Putu Wira Buana, I Nyoman Piarsa
Department of Information Technology, Udayana University, Indonesia

Article Info

Article history:

Received Feb 12, 2021
Revised May 20, 2021
Accepted Jun 1, 2021

Keywords:

Deep learning
Face identification
K-nearest neighbor
Random forest
Support vector machine

ABSTRACT

The use of a face as a biometric to identify a person in order to keep the system safe from an unauthorized person has advantages over other biometric characteristics. The face as a biometric has more structure and a wider area than other biometrics, while can be retrieved in a non-invasive manner. We proposed a cloud-based architecture for face identification with deep learning using convolutional neural network. Face identification in this study used a cloud-based engine with four stages, namely face detection with histogram of oriented gradients (HOG), image enhancement, feature extraction using convolutional neural network, and classification using k-nearest neighbor (KNN), SVM, as well as random forest algorithm. This study conducted a classification experiment with cloud-based architecture using three different datasets, namely Faces94, Faces96 and University of Manchester Institute of Science and Technology (UMIST) face dataset. The results from this study are with the proposed cloud-based architecture, the best accuracy is obtained by KNN algorithm with an accuracy of 99% on Faces94 dataset, 99% accuracy on Faces96 dataset, 97% on UMIST face dataset, and performance of the three algorithms decreased in UMIST face dataset with facial variations from various angles from left to right profile.

Corresponding Author:

Aditya Herlambang
Department of Information Technology
Udayana University
Raya Kampus Unud Jimbaran St., Kuta Selatan, Kabupaten Badung, Bali, Indonesia
Email: herlambangadt@gmail.com

1. INTRODUCTION

Facial biometrics have many distinct advantages compared to other biometric characteristics [1]. The camera can be used to peform non-invasive way to capture faces. Face also has a richer texture and a wide area. It became an important method of biological authentication and attracted interest in different domains [2]. There are several methods that can be applied before identifying faces, one of which is detecting faces in the image that can be done using histogram of oriented gradients (HOG). Based on the first study that proposed the HOG method, after evaluating current descriptors based on edge and gradient, it was experimentally shown that the grids of the HOG descriptors outperform the array of features usable for human detection by a wide margin [3]. For person-dependent and person-independent versions, Tambi *et al.* [4] created an improved face recognition system by detecting face using HOG and convolutional neural network (CNN) for extracting relevant facial features. Their research resulted in 96.19% accurate result for Yale dataset.

Deep learning have distinguished classification and learning performance, where it can automatically extracts low and high level features for classification [5]. In deep learning, facial feature can be extracted by single CNN architecture from large amount of image containing faces. CNN is intended to

process data that has a known network such as a topology. CNN is generally used to identify image characteristics and trends in time series images. CNN involves multiple connections. Convolution, pooling, and fully-connected layers are the building blocks or layers of CNN architecture [6]. As in Ding and Tao [7] CNN sets are used to extract facial characteristics from multimodal information. A verification rate of 98.43% and recognition rate of 99.0% achieved on the labeled faces in the wild (LFW) database. Other example of facial feature extraction is in Widiakumara *et al.* [8] where this study resulted in a face identification application using Android-based Eigenface with a trial success of 68% and a false positive rate of 32%.

Classification using KNN can be done after feature extraction to identify face. As in Wirdiani *et al.* [9] carried out three stages to identify face including face detection, feature extraction and classification. The method used to extract features is principal component analysis (PCA) and the method to perform classification is k-nearest neighbor (KNN). This paper produced a program using Python programming language to identify faces. The result obtained from several test of k values gives the best accuracy of 81% with $k = 1$ and the greater k value gives smaller accuracy. Apart from the KNN, the classification of the data from facial feature extraction can also be done using the support vector machine (SVM) algorithm as in Senthilkumar and Gnanamurthy [10]. This research used the SVM classifier to compare performance changes in the recognition rate of different facial recognition techniques. SVM resulted in better classification compared to other methods. Another classification algorithm that can be used for classification from facial feature result is random forest algorithm. As in Mady and Hilles [11] Random Forest classifier is used to classify facial feature extracted using histogram of oriented gradients (HOG) and local binary pattern (LBP). This study resulted in 97.6% recognition accuracy on Mediu staff database.

Face Identification with deep learning using Convolutional Neural Network in this study will be running in a cloud-based architecture using Flask Framework so that image can be processed immediately after received by the cloud server, and then perform HOG for face detection before doing the image enhancement process, feature extraction with the CNN to produce 128-d embeddings, then performs classification algorithm comparisons between KNN, Linear SVM and Random Forest to find the best algorithm in terms of accuracy to classify the 128-d embeddings generated by deep learning using CNN.

2. RESEARCH METHOD

Cloud-based architecture for face identification with deep learning consist of two phases, namely training and testing stage. Training stage aims to produce a classification model that will be used in the testing phase and store it in the classification model database, and save the results of facial feature extraction to the facial feature database in the cloud. At the training stage, the preprocessing stage will be carried out after the device sends train images, then face detection is conducted with HOG, feature extraction using CNN, then the model is trained with KNN. Figure 1 shows the training stage with cloud-based architecture. The second stage of facial identification after the training stage is testing stage. Test image will be received by the Flask Framework in the cloud. After the test image is received, the testing process will be carried out immediately. Figure 2 shows the testing phase with cloud-based architecture.

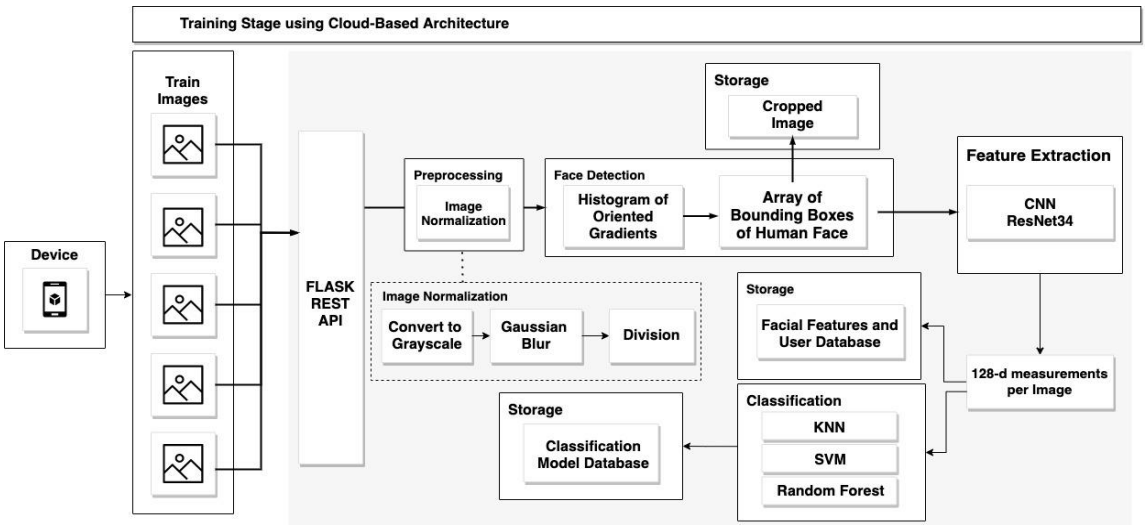

Figure 1. Training stage of face identification

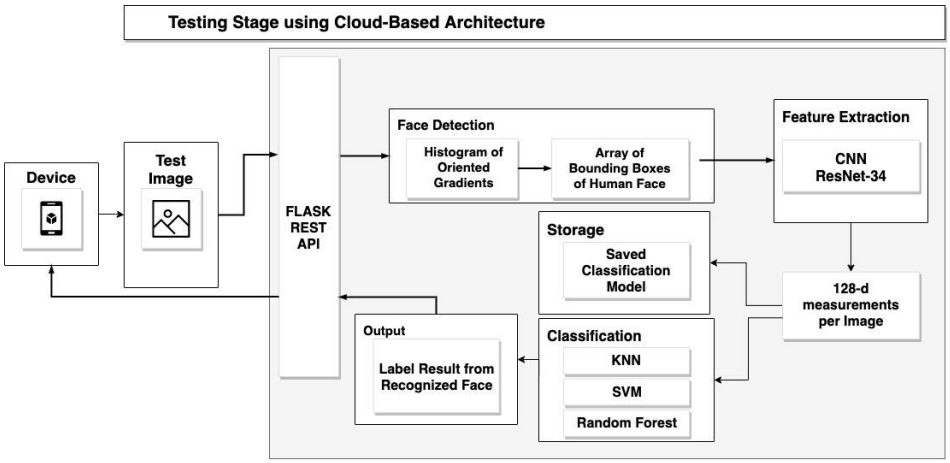

Figure 2. Testing stage of face identification

2.1. Face image dataset

There are 3 separate datasets, namely the University of Manchester Institute of Science and Technology (UMIST) face dataset, Faces94 and Faces96, used for the training and testing phases. The UMIST face dataset contains 575 faces from 152 subjects taken from left profile to right angle [12]. The UMIST face dataset example can be seen in Figure 3. Faces94 comprises 3060 faces from 153 subjects who are seated at around the same distance from the camera and asked to talk as twenty consecutive photographs are taken. The speech is used to incorporate changes in facial expressions that are mild and normal. The Faces94 Dataset example can be seen in Figure 4. Faces96 includes 3040 faces from 152 subjects taken using a fixed camera, where the subject takes one or more steps forward towards the camera when the picture is taken to introduce major head differences between images of the same person [13]. The Faces96 Dataset example can be seen in Figure 5.

Figure 3. UMIST face dataset example

Figure 4. Faces94 dataset example

Figure 5. Faces96 dataset example

2.2. Face image dataset

Image enhancement's main goal is to improve graphical clarity and accuracy of an image, or to give an enhanced picture of transformation for upcoming image processing [14]. The image enhancement stage in this study consists of conversion to grayscale, image smoothing with a Gaussian Filter, and dividing gray by morphology image. Noise in the image is reduced by doing image smoothing with gaussian filter, thereby improving the quality of an image [15]. We use image enhancement feature from OpenCV because it has the benefit of being a multi-platform library [16] so our cloud-based architecture is not limited to a specific platform. Figure 6 is the result of the image enhancement carried out in this study.

Figure 6. Image enhancement

2.3. Face detection

Face detection is the process of discovering bounding boxes of human face in an image sequence. This research makes use of the HOG method and SVM to detect face in an image. With HOG, the picture pixel's horizontal gradient and the picture pixel's vertical gradient is presented in (1) and (2). The gradient magnitude and pixel direction can be seen in (3) and (4).

$$G_a(a,b) = H(a+1,b) - H(a-1,b) \quad (1)$$

$$G_b(a,b) = H(a,b+1) - H(a,b-1) \quad (2)$$

$$G(a,b) = \sqrt{G_a(a,b)^2 + G_b(a,b)^2} \quad (3)$$

$$a(a,b) = \tan^{-1}\left(\frac{G_b(a,b)}{G_a(a,b)}\right) \quad (4)$$

The gradient has a magnitude and a direction at each pixel. In (3) is used to measure the gradient direction, while (4) is used to calculate the gradient magnitude. The maximum of the three channels' gradients is the magnitude of the gradient at a pixel, and the angle is the angle equivalent to the maximum gradient. To count the gradient direction using (3) and gradient magnitude using (4), the provided frame is divided into cells, which are pixel-sized rectangular or circular areas. For each cell, the gradient feature vectors are then calculated. In each single frame, the feature vector is then constructed using this gradient feature vector. Finally, the HOG feature vector is produced by combining all gradient feature vectors derived from different images, and then inputted to the SVM to extract an array of bounding boxes for the human face [17]. Input image and feature vectors generated from the HOG method can be seen in Figure 7.

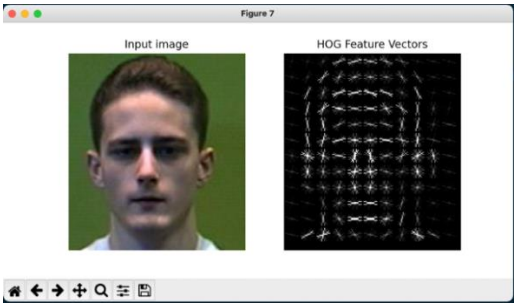

Figure 7. Input image and HOG feature vectors result

2.4. Feature extraction using CNN

Feature extraction is a phase for extracting facial characteristic features from an image. CNN from Dlib and ResNet architecture with 29 Conv layers, which is a variety of ResNet-34 using fewer layers and half the number of filters in each layer [18] are used in this study to extract features. ResNet allows deeper architectural training, because the layer learns residual functions by referring to layer inputs and does not learn functions that are not referenced. This allows the network to be resilient to the gradient disappearing problem and handle the drop in accuracy that occurs in conventional deep grids [19]. This study used CNN architecture from Dlib [20] to extract facial features. The CNN architecture which combines local receptive fields, shared weights and pooling [21] used in this study can be seen in Figure 8. This method generates 128-d feature vectors from facial images that have been detected in the face detection stage. Figure 9 is an example of the feature extraction results from one of the faces in the dataset.

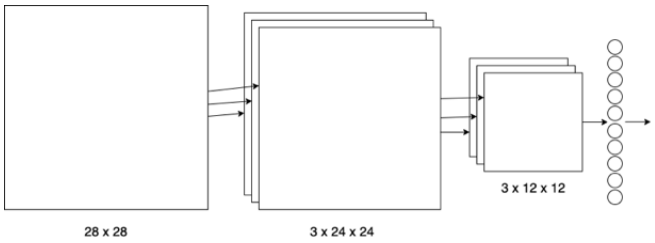

Figure 8. CNN architecture

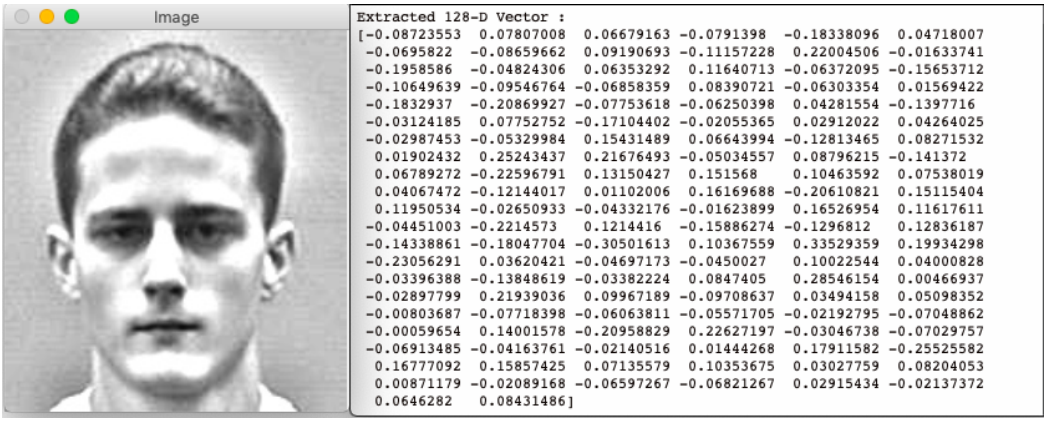

Figure 9. Feature extraction using CNN

2.5. Classification with KNN algorithm

One of the simple algorithms that can be used in the classification process to match data between testing and training data from face datasets is KNN [9]. KNN was used in the early 1970s for statistical estimation and pattern recognition [22]. KNN performs well on many samples. When a new test sample appears, the distance between it and other trained samples will be determined using the k value, and the test sample will be calculated by the class member whose sample is nearest to the test sample [23]. This study used k=1 to produce prediction result based on the nearest neighbor. This study use classification function from Scikit-learn [24]. To find the nearest neighbor, this study used Euclidean distance formula that can be seen in (5).

$$d(x, x_i) = \sqrt{\sum_{j=1}^{n} (x_j - x_{ij})^2} \quad (5)$$

The (5) is the euclidean distance formula where x is the parameter of testing data, xi is the parameter of training data. The parameter n is the dimension of the feature vector. Euclidean Distance is utilized to increase accuracy by measuring the distance between points along a straight line, and is teaching and

research results. This distance method uses the pythagorean theorem. Calculation with euclidean distance aims to compare the minimum distance of the training image and the test image.

2.6. Classification with SVM algorithm

Support Vector Machine (SVM) is a classification algorithm that study how to label objects by using examples. SVM is a mathematical object that optimizes a mathematical function in relation to a given data set [25]. Between two groups of results, the separating hyperplane with the largest margin is found by SVM [26]. The kernel used in this study is Linear which the function can be seen in (6).

$$K(x_i, x_j) = x_i^T x_j \quad (6)$$

2.7. Classification with random forest algorithm

Random forest is collection of tree predictors put together in a random order [27]. Each tree in the forest is based on a random vector value that it samples independently and with the same distribution. For an ensemble of classifiers $h_1(x), h_2(x), \ldots, h_k(x)$, the margin function can be used in an equation with a training range randomly selected from a distribution of random vectors X, Y. The margin function can be seen in (7).

$$mg(X,Y) = av_k I(h_k(X) = Y) - max_{j \neq Y} av_k I(h_k(X) = j) \quad (7)$$

The margin indicates how much the right class's average number of votes in X, Y exceeds the other classes' average number of votes. The smaller the margin of error, the more confident the classification [27]. Random Forest block diagram of random vector and the results are shown in Figure 10.

The facial feature dataset is used as training data for different decision trees. This dataset includes observations and attributes that will be picked at random as nodes are separated. Each tree's leaf node represents the final output generated by that particular decision tree. The end product is chosen by a majority-voting method. In this case, the final output of this method is the output selected by the plurality of decision trees.

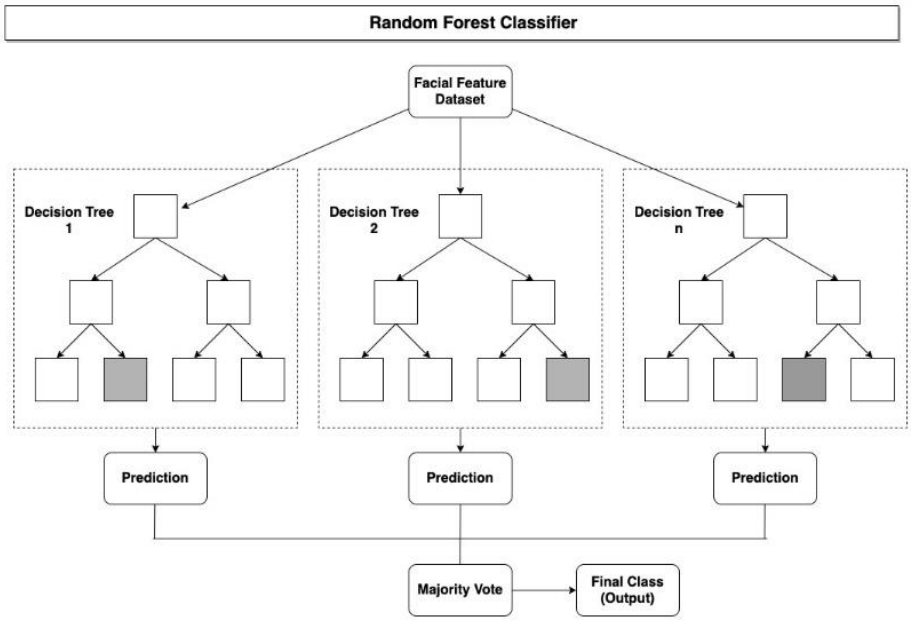

Figure 10. Random forest classifier block diagram

3. RESULTS AND DISCUSSION

In this study, three different datasets, namely Faces94, Faces96, and UMIST face dataset, were used to calculate the accuracy of the identification results. The purpose of using three different datasets was to calculate the accuracy of face identification from different conditions. Table 1 shows the dataset used in this analysis.

Faces94 is used to calculate accuracy for moderate and natural variations in facial expressions with the speaking subject, Faces96 is used to calculate accuracy for variations in head movement at different distances from the camera, and the UMIST face dataset is used to calculate accuracy for facial variations from various angles from left to right profile. Both Faces94 and Faces96 are all frontal view while UMIST face dataset captured from different angle from left to right profile.

Table 1. Training and testing data

Dataset	Training data	Testing data	Total class
Faces94	2142	918	153
Faces96	2189	905	152
UMIST Face Dataset	402	173	20

For k-nearest neighbor, linear SVM, and random forest (RF) classification, this study used Scikit-learn library. The k value parameter used in this study is $k = 1$ to produce prediction result based on the nearest neighbor. The result of precision, recall, F1-score, support and processing time can be seen in Tables 2-4.

Table 2. Precision, recall, F1-score, support and processing time for Faces94 dataset

Algorithm	Precision	Recall	F1-Score	Support	Processing time (s)
KNN	99%	99%	99%	918	0.76
Linear SVM	98%	99%	99%	918	1.42
Random Forest	98%	98%	98%	918	0.59

Table 3. Precision, recall, F1-score, support and processing time for Faces96 dataset

Algorithm	Precision	Recall	F1-Score	Support	Processing time (s)
KNN	99%	99%	99%	905	0.92
Linear SVM	98%	98%	98%	905	1.40
Random Forest	97%	98%	97%	905	0.90

Table 4. Precision, recall, F1-score, support and processing time for UMIST face dataset

Algorithm	Precision	Recall	F1-Score	Support	Processing time (s)
KNN	97%	98%	97%	173	0.06
Linear SVM	85%	74%	76%	173	0.07
Random Forest	93%	93%	92%	173	0.05

From Table 2 to Table 4, it is shown that the three algorithms have relatively good performance with F1-score > 95% on the Faces94 and Faces96 datasets. However, the three algorithms experienced a decrease in F1-score on the UMIST face dataset. The cause of the decrease in the F1-score on the UMIST face dataset is the variation in taking facial images from the left side to the right side on the UMIST face dataset, which is not available in the Faces94 and Faces96 datasets. In terms of recognition time, random forest yields the fastest identification time among the other 2 algorithms.

Among the three algorithms, KNN has a stable performance compared to linear SVM and random forest. On the UMIST face dataset, the KNN F1-score was only reduced by 2% from the highest F1-score on the Faces96 dataset, namely 99%, the SVM linear F1-score was 23% less than the highest F1-score on Faces94, and the random forest F1-score was reduced by 6% of the highest F1-score on the Faces94 dataset. From the three tests conducted, KNN obtained the highest F1-score that is 99% on the Faces94 dataset, 99% on the Faces96 dataset, and 97% on the UMIST face dataset.

Apart from the F1-score, evaluation of the three algorithms can also be done by looking at the false acceptance rate (FAR) and false rejection rate (FRR). FAR is the number of times that unwanted individuals or impostors are wrongly accepted during recognition, or the percentage of times that false acceptance happens. FRR is the number of identification instances in which authorized individuals or true label was wrongly rejected. The comparison between accuracy, FAR, and FRR for each algorithm can be seen in the graph in Figures 11-13.

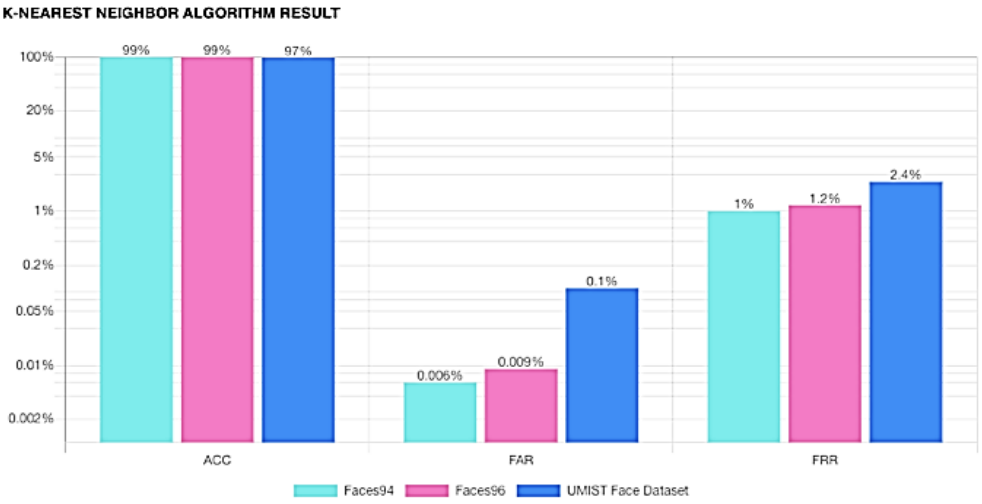

Figure 11. Accuracy, FAR, and FRR of KNN algorithm

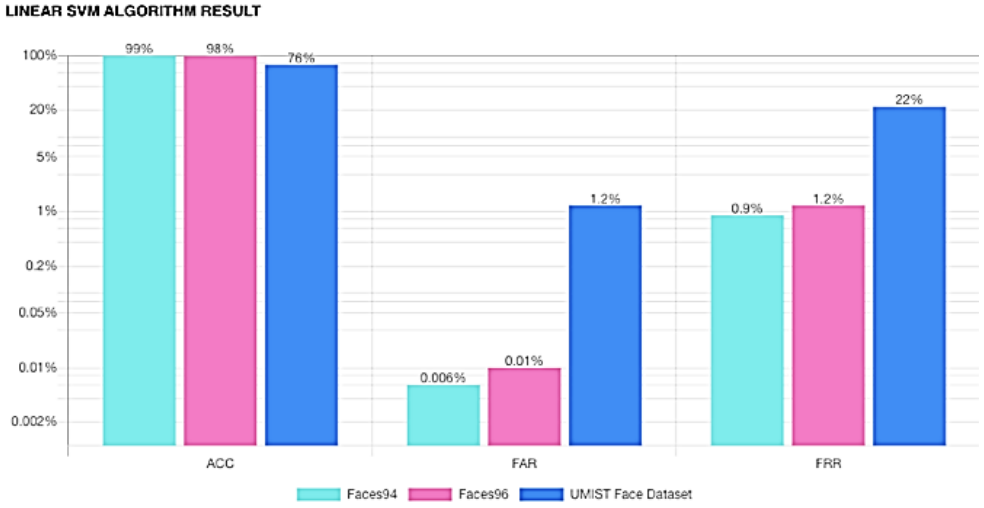

Figure 12. Accuracy, FAR, and FRR of linear SVM algorithm

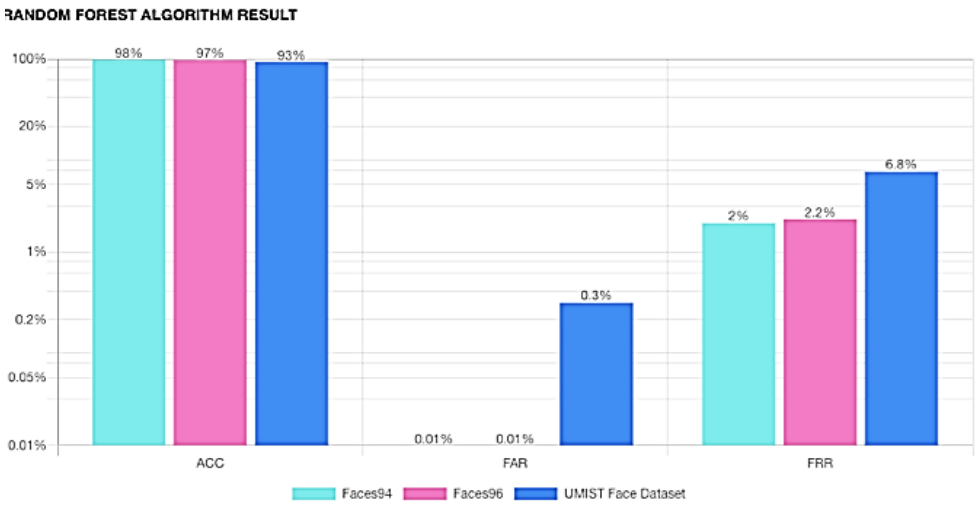

Figure 13. Accuracy, FAR, and FRR of random forest algorithm

From Figure 11 through Figure 13, it can be seen that there is an increase in FAR and FRR and a decrease in accuracy on the UMIST face dataset. The increase in FAR and FRR on the UMIST face dataset indicates that the three classification algorithms are more vulnerable to false acceptance and false rejection on the face dataset taken from the left profile to the right profile than the dataset taken from the front side of the face. The smaller the FAR and FRR values, the higher the reliability of the classification algorithm because it indicates the fewer false acceptance and false rejection percentages. For each algorithm, the lowest FAR are 0.006% for KNN and Linear SVM, and 0.01% for random forest. For each algorithm, the lowest FRR are 1% for KNN, 0.9% for Linear SVM, and 2% for random forest.

4. CONCLUSION

In this study, we conduct an experiment in cloud-based architecture using classification algorithms on face identification with CNN to extract facial feature from image and produces 128-d embeddings. The data sources came from three different datasets, namely Faces94, Faces96 and UMIST face dataset with different characteristics. This study uses four stages to identify faces, namely image enhancement, face detection, feature extraction, and classification using the KNN algorithm, linear SVM, and random forest. From the classification results, the three algorithms have decreased accuracy on the UMIST face dataset which has the characteristics of image captured from the left side to the right side. From the result of this study, it is concluded that with the proposed cloud-based architecture, the best accuracy is obtained by KNN algorithm with an accuracy of 99% for the Faces94, 99% accuracy for Faces96, and 97% accuracy for UMIST face dataset.

ACKNOWLEDGEMENTS

We would like to express our gratitude to Udayana University Department of Information Technology for providing sufficient facilities for the completion of this study.

REFERENCES

[1] R. Blanco-Gonzalo, N. Poh, R. Wong, and R. Sanchez-Reillo, "Time evolution of face recognition in accessible scenarios," *Human-centric Computing and Information Sciences*, vol. 5, no. 1, pp. 0–11, 2015, doi: 10.1186/s13673-015-0043-0.

[2] C. Li, W. Wei, J. Li, and W. Song, "A cloud-based monitoring system via face recognition using Gabor and CS-LBP features," *J. Supercomput.*, vol. 73, no. 4, pp. 1532–1546, 2017, doi: 10.1007/s11227-016-1840-6.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings-2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, vol. I, 2005, pp. 886–893, doi: 10.1109/CVPR.2005.177.

[4] P. Tambi, S. Jain, and D. K. Mishra, *Person-Dependent Face Recognition Using Histogram of Oriented Gradients (HOG) and Convolution Neural Network (CNN)*, vol. 870, Springer Singapore, 2019.

[5] L. Arnold, S. Rebecchi, S. Chevallier, and H. Paugam-Moisy, "An introduction to deep learning," *ESANN 2011 - 19th Eur. Symp. Artif. Neural Networks*, pp. 477–488, 2011, doi: 10.1201/9780429096280-14.

[6] I. Namatēvs, "Deep Convolutional Neural Networks: Structure, Feature Extraction and Training," *Inf. Technol. Manag. Sci.*, vol. 20, no. 1, pp. 40–47, 2018, doi: 10.1515/itms-2017-0007.

[7] C. Ding and D. Tao, "Robust Face Recognition via Multimodal Deep Face Representation," *IEEE Trans. Multimed.*, vol. 17, no. 11, pp. 2049–2058, 2015, doi: 10.1109/TMM.2015.2477042.

[8] N. H. Barnouti, S. S. M. Al-Dabbagh, and M. H. J. Al-Bamarni, "Real-Time Face Detection and Recognition Using Principal Component Analysis (PCA) – Back Propagation Neural Network (BPNN) and Radial Basis Function (RBF)," *Journal of Theoretical and Applied Information Technology*, vol. 91, no. 1, pp. 28–34, 2016.

[9] N. K. A. Wirdiani, P. Hridayami, N. P. A. Widiari, K. D. Rismawan, P. B. Candradinata, and I. P. D. Jayantha, "Face Identification Based on K-Nearest Neighbor," *Sci. J. Informatics*, vol. 6, no. 2, pp. 150–159, 2019, doi: 10.15294/sji.v6i2.19503.

[10] R. Senthilkumar and R. K. Gnanamurthy, "Performance improvement in classification rate of appearance based statistical face recognition methods using SVM classifier," *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2017, pp. 1-7, doi: 10.1109/ICACCS.2017.8014584.

[11] H. Mady and S. M. S. Hilles, "Face recognition and detection using Random forest and combination of LBP and HOG features," *2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*, 2018, pp. 1-7, doi: 10.1109/ICSCEE.2018.8538377.

[12] D. B. Graham and N. M. Allinson, "Characterising Virtual Eigensignatures for General Purpose Face Recognition," in *Face Recognition: From Theory to Applications*, H. Wechsler, P. J. Phillips, V. Bruce, F. F. Soulié, and T. S. Huang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 446–456, 1998.

[13] D. Hond and L. Spacek, "Distinctive descriptions for face processing," in *Proceedings of the 8th British Machine Vision Conference BMVC97, Colchester, England*, 1997, pp. 320–329.

[14] P. Janani, J. Premaladha, and K. S. Ravichandran, "Image Enhancement Techniques: A Study," *Indian Journal of Science and Technology*, vol. 8, no. 22, pp. 83–89, Sep. 2015, doi: 10.17485/ijst/2015/v8i22/79318.
[15] I. Agustina, F. Nasir, and A. Setiawan, "The Implementation of Image Smoothing to Reduce Noise using Gaussian Filter," *International Journal of Computer Applications*, vol. 177, no. 5, pp. 15–19, 2017, doi: 10.5120/ijca2017915755.
[16] S. Emami and V. P. Suciu, "Facial Recognition using OpenCV," *J. Mobile, Embed. Distrib. Syst.*, vol. 4, no. 1, pp. 38–43, 2012.
[17] A. Adouani, W. M. Ben Henia and Z. Lachiri, "Comparison of Haar-like, HOG and LBP approaches for face detection in video sequences," *2019 16th International Multi-Conference on Systems, Signals & Devices (SSD)*, 2019, pp. 266-271, doi: 10.1109/SSD.2019.8893214.
[18] D. E. King, "Dlib-ml: A Machine Learning Toolkit," *J. Mach. Learn. Res.*, vol. 10, no. 10, pp. 1755–1758, 2009, doi: 10.1145/1577069.1755843.
[19] P. Korfiatis, T. L. Kline, D. H. Lachance, I. F. Parney, J. C. Buckner, and B. J. Erickson, "Residual Deep Convolutional Neural Network Predicts MGMT Methylation Status," *J. Digit. Imaging*, vol. 30, no. 5, pp. 622–628, 2017, doi: 10.1007/s10278-017-0009-z.
[20] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 07-12-June, 2015, pp. 815–823, doi: 10.1109/CVPR.2015.7298682.
[21] S. Sharma, K. Shanmugasundaram and S. K. Ramasamy, "FAREC — CNN based efficient face recognition technique using Dlib," *2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, 2016, pp. 192-195, doi: 10.1109/ICACCCT.2016.7831628.
[22] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, John Willey & Sons, 2012.
[23] F. Mahmud, B. Islam, A. Hossain and P. B. Goala, "Facial Region Segmentation Based Emotion Recognition Using K-Nearest Neighbors," *2018 International Conference on Innovation in Engineering and Technology (ICIET)*, 2018, pp. 1-5, doi: 10.1109/CIET.2018.8660900.
[24] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[25] W. S. Noble, "What is a support vector machine?," *Nat. Biotechnol.*, vol. 24, no. 12, pp. 1565–1567, 2006, doi: 10.1038/nbt1206-1565.
[26] Y. Chang and C. Lin, "Feature Ranking Using Linear SVM," *Featur. Rank. Using Linear SVM*, vol. 3, pp. 53–64, 2008.
[27] R. A. Nugrahaeni and K. Mutijarsa, "Comparative analysis of machine learning KNN, SVM, and random forests algorithm for facial expression classification," *2016 International Seminar on Application for Technology of Information and Communication (ISemantic)*, 2016, pp. 163-168, doi: 10.1109/ISEMANTIC.2016.7873831.