Design of Implementation of Deep Learning Based Face Recognition System

Adi Irwanto 1[1]*, Siti Fatimah Zuhro 2[2], Sapriyadi 3[3] Fery Updi 4[4], Purwani Husodo 5[5]

[12345]Computer Science Study Program/Department, Faculty of Engineering, Muhammadiyah University AR Fachruddin

Abstract: This study discusses the face recognition system. In general, the face recognition system is divided into two stages, namelynamely system face detection, which is a pre-processing step followed by a face recognition system. This step will be done quickly by humans but takes a long time for computers. This human ability is what researchers have wanted to emulate in recent years as biometric technology in computer vision to create face recognition models on computers. Deep learning is in the spotlight in the development of machine learning, the reason being that deep learning has achieved extraordinary results in computer vision. Based on this, the author has the idea to create a face recognition system by implementing deep learning using the CNN method and implementing it in the openFace library. The CNN method is still superior and widely used because it has good accuracy. The initial process is taking pictures of the face to be photographed as a data set. From the dataset, face preprocessing will be carried out, namely to extract the face vector features into 128-d and classify the face vector. The contribution of this research is the addition of features to improve the accuracy of the face recognition system using the CNN method. The results of this study obtained a precision value of 98.4%, a recall of 98% and an accuracy of 99.84%.

Keywords: *Deep Learning, Face Recognition, Convolution Neural networks.*

INTRODUCTION

Facial recognition systems have become a frequently studied topic in the field of computer vision in recent decades. This system has been applied in several fields, for example, smartphones for facelock, immigration and also in social media for facial tagging features [1]. Facial recognition systems are generally divided into two stages, namely the facial detection system which is the initial stage (pre-processing) and continued with the facial recognition system [2]. This stage is very fast for humans but takes a long time for computers. This human ability is what researchers have wanted to duplicate in recent years as biometric technology in the field of computer vision. The purpose of this duplication is to form a facial image recognition model on

a computer. Lately, Deep Learning has been in the spotlight in the development of Machine Learning. The reason is because Deep Learning has achieved extraordinary results in computer vision.

Deep Learning is a learning method that is carried out by machines to be able to understand and classify an object, namely the main one in the application that is built is the face that is captured in the form of an image [3]. Convolutional Neural Network (CNN) is one of the methods in Deep Learning that is made to cover the weaknesses of the previous method. There are several weaknesses in the previous method, but with this model a number of free parameters can be reduced and deformation of the input image such as translation, rotation and scale can be handled.

Various studies have begun to test several face recognition methods to find a good method with high recognition accuracy. One of them was conducted by Arsal in 2020, in this study the experimental results showed an accuracy level of 95% using deep learning with the Convolutional Neural Network method [4].

Research conducted by Achmad in 2019, in this study proposed a classification of emotions based on facial features to help improve the quality of consumer testing. The method used in this study is Convolutional Neural Network (CNN). The test results show that certain learning rate values can train the architecture better than other learning rate values. The best accuracy value in this study was 86.4% and the average accuracy was 80.7% [5]. Research conducted by Meenakshi in 2019, in this study proposed a facial recognition system application using the Convolutional Neural Network (CNN) method.

The proposed facial recognition system is implemented and analyzed in the MATLAB 2018 platform. The accuracy results of this study are 98.75% [6]. Along with the many developments and research on Deep Learning, many libraries have emerged with a focus on studying artificial neural networks. One example is OpenFace. OpenFace is a library used for face recognition purposes and is based on the neural network design of DeepFace and GoogleNet with some modifications. From previous research, the author designed a facial recognition system using the Convolutional Neural Network method by applying the OpenFace library to deepen knowledge about deep learning and determine the level of accuracy produced.

Face Recognition

Facial recognition is one of the biometric technologies that has been widely applied in security systems besides retinal recognition, fingerprint recognition and iris recognition. In its application, facial recognition uses a camera to capture a person's face and then compares it with a face that has previously been stored in a specific database[7].

Face recognition involves many variables, such as source images, processed imagery, extracted images, and a person's profile data. It also requires a sensing device in the form of a

camera sensor and a method to determine whether the image captured by the webcam is a human face or not, as well as to determine the profile information that matches the intended facial image.

Convolutional Neural Network

Convolutional Neural Network (CNN) is one of the classification methods included in the deep learning group that uses a convolution layer to convolve an input with a filter. CNN consists of two main stages, namely feature learning and classification. The feature learning stage consists of a convolution layer, ReLU (activation function) and pooling layer while the classification stage consists of flatten, fully-connected layer, and prediction. In each part of CNN there are two main processes, namely feedforward and backpropagation[5].

Deep Learning Cafe

CAFFE (Convolutional Architecture for Fast Feature Embedding) is one of the many deep learning frameworks that can be used to design and run neural network classifiers. Caffe uses Google's prototxt format for neural network specifications. Each layer is defined separately and the input and output are multidimensional arrays called blobs. The layers are stacked vertically with the input at the bottom and the output at the top. Each layer expects a certain number of bottom and top blobs with a certain dimension. Chaining these blobs and layers produces a deep neural network [8].

OpenFace

OpenFace is a library used for face recognition purposes and is based on the neural network design of DeepFace and GoogleNet with some modifications. OpenFace uses dlib 22 to detect facial regions in images and generates a box around each face that can be under different poses.

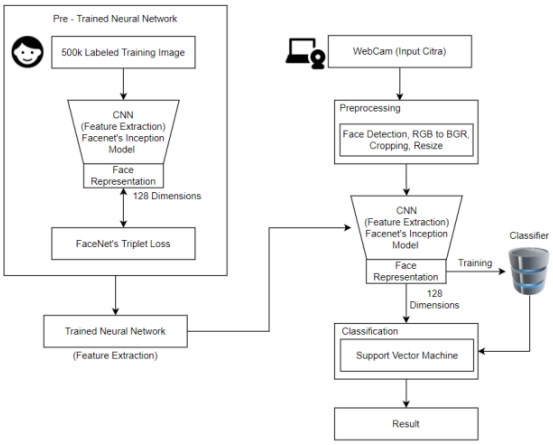
OpenFace uses a 2D affine transformation as its preprocessing method, which adjusts the nose and eye angles relatively close to the average location by resizing and cropping the image to the edges of landmarks generated by the dlib face detector. The result of this transformation is a normalized image of 96 x 96 pixels.

RESEARCH METHODS

In this study, the CNN method used for face recognition consists of several stages. These stages begin with the collection of training data used for neural network learning. The Convolutional Neural Network (CNN) method used in this study has a pre-trained neural network as a feature extraction method. Training facial image data is used to train the neural network.

Training consists of several processes. The first process is face detection in the image. When there is a face image, a feature extraction process is carried out which will produce the characteristic values of each face. These characteristic values are stored and used as an identifier

(classifier). When testing is carried out, the classifier will be used for face classification. Each process carried out will be discussed in more detail in the next section. The design of the research method is shown in Figure 1, where the workflow of this system consists of several stages.



Gambar 1. Proses sistem pengenalan wajah

Pre-Trained Neural Network

In this study, a pre-trained neural network model is used, namely a neural network that has been trained using many facial datasets. The pre-trained neural network used uses the openface nn4.small.2 model which has been trained on 500 thousand training image datasets. The pre-trained neural network process works with several training data, namely training facial images of familiar people, training facial images with the same person and facial images of different people. The three image data are then processed to produce feature values for each image using the Convolutional Neural Network (CNN) algorithm.

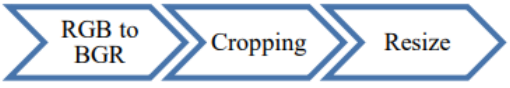
This Neural Network is then retrained to ensure that 2 images with the same person have identical feature values while having different values for different people. This process is repeated for other training data until the ideal neural network condition is obtained. The results of the Neural Network training are used as the Pre-Trained Neural Network for the next facial recognition process.

Image Acquisition / Image Input

The first process to be carried out is the image acquisition stage, which functions to capture facial images using a webcam.

Preprocessing

The following are the steps in the preprocessing stage.



Gambar 2. Langkah – langkah *PreProcessing*

Feature Extraction

Feature extraction is a process that functions to obtain effective and useful information to distinguish different people's faces from aligned facial images [9]. After face preprocessing is carried out by detecting, cropping and resizing the face. The method for feature extraction uses Convolutional Neural Network (CNN). The CNN method that has used Pre-Trained Neural Network to represent human faces by producing 128 facial feature values [10]. The facial feature extraction process using the Convolutional Neural Network method has several stages of deep neural network layers. These stages start from the normalized 96x96x3 image input model. The layers used in feature extraction only reach the inception layer with final pooling which produces 128 feature values. The complete Convolutional neural network architecture can be seen in table 1.

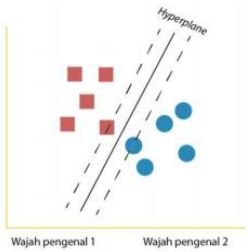
Tabel 1. Convolutional Neural Network nn4.small2 Openface

Type	Output Size	#1x1	#3x3 reduce	#3x3	#5x5 reduce	#5x5	Pool proj
conv1 (7 x 7 x 3,2)	48 x 48 x 64						
max pool + norm	24 x 24 x 64						m 3 x 3,2
inception (2)	24 x 24 x 192		64	64			
norm + max pool	12 x 12 x 192						m 3 x 3,2
inception (3a)	12 x 12 x 256	64	96	96	16	32	m, 32p
inception (3b)	12 x 12 x 320	64	96	96	32	64	ℓ_2, 64p
inception (3c)	6 x 6 x 640		128	128	32	64,2	m 3 x 3,2
inception (4a)	6 x 6 x 640	256	96	96	32	64	ℓ_2, 128p
inception (4e)	3 x 3 x 1024		160	160	64	128,2	m 3 x 3,2
inception (5a)	3 x 3 x 736	256	96	96			ℓ_2, 96p
inception (5b)	3 x 3 x 736	256	96	96			m, 96p
avg pool	736						
Linear	128						
ℓ_2 normalization	128						

Based on table 1, an image extraction architecture using the nn4.small2 Openface Convolutional Neural Network model is described. Each row in the table shows the Convolutional Neural Network layer and each column shows the filtering pooling and inception layer parameters. In the convolution process conv1, the image input is processed to produce an output with a size of 48x48x64. The output from conv1 is then pooled and normalized using the kernel pool proj m 3 x 3.2. The pooling process produces an output of 24x24x192. The results of this output are used in the inception process using 64 kernels #3x3 reduce and 64 kernels #3x3. The output from the inception process is normalized again with max-pool proj m 3x3.2. This output is used in several inception layers with several filters to produce a final output of 736 values and is again avg pooled and normalized to produce 128 extraction values. Convolutional Neural Network with 128 measurement values is generated for each face and will be stored in the database. The value data for a recognition is trained using the Support Vector Machine model.

Classification

Facial image classification is done to find out whose face is in the image. To compare facial values, the Support Vector Machine (SVM) classification algorithm is used. Support Vector Machine (SVM) is a classification method first introduced by Vapnik in 1998[11]. The classification process is carried out by comparing the image values in the database (classifier) with the input image data being tested. Similar feature values will be classified into the same class. The trained image has a boundary called a hyperplane for each identifier. This hyperplane separates the facial feature values from one another. The hyperplane is obtained by placing a boundary from the furthest value distance of 2 different classes. Figure 4 shows the hyperplane of 2 different facial classes.



Gambar 4. Hyperplane SVM

The SVM method used is Linear kernel SVM. Linear kernel is the simplest kernel function. Linear kernel is used when the analyzed data is already linearly separated. Linear kernel is suitable when there are many features because mapping to a higher dimensional space does not really improve performance as in text classification. This function is good to use when the data is already linearly separated, this kernel is done with equation 1 [12] where K (xi, xj) is the kernel function while xi and xj are pairs of two training data. K (xi, xj) = xi . xj (1) In conducting analysis with this linear kernel SVM function, to obtain optimal accuracy, the C (Cost) parameter value will be sought. The C parameter used for the first is with a value of C = 0.1, then this parameter value is enlarged to find the most optimal accuracy in Linear kernel SVM.

Recognition Result

After going through the pre-processing process, feature extraction is then carried out in the facial identification process, when a user's facial image matches the image data in the database, it will display the identity of the user who owns the facial image.

RESULTS AND DISCUSSION

System Testing

This section explains the system testing using Convolutional Neural Network (CNN) using the test dataset. In the first stage, the training data image must go through the stages of face detection, image conversion, and cropping. The results of face detection and cropping are shown in Figure

5.

Gambar 5. Hasil Cropping Image

After preprocessing is done, then feature extraction is done. In this study, there are 4 feature extractions used, namely:

a. ROI

The ROI process in this study is used to limit or reduce the processing area. The limitation of the processing area is done by determining the face area. The limitation of the face area is done so that objects outside the area do not add noise to the processed image. The purpose of implementing ROI in this study is to increase processing time.

b. Bounding Box

A bounding box is an imaginary box that surrounds an identified object. The bounding box itself is in the form of a box whose size is the same as the size of the identified object. In the bounding box process, input data is obtained from the results of the face detection process, then the input is processed to find the values of X, Y, W, H. Then when the values of X, Y, W, and H have been found, the next step is to create a bounding rectangle with these values. This feature is used to reduce the high processing time. Because it is able to mark certain areas so that it can be used to optimize system performance to detect, calculate and classify faces in real time. If not, processing is carried out on all image pixels without exception. The following is a snippet of the extraction process program, which is then continued by the embedding process.

Potongan Program

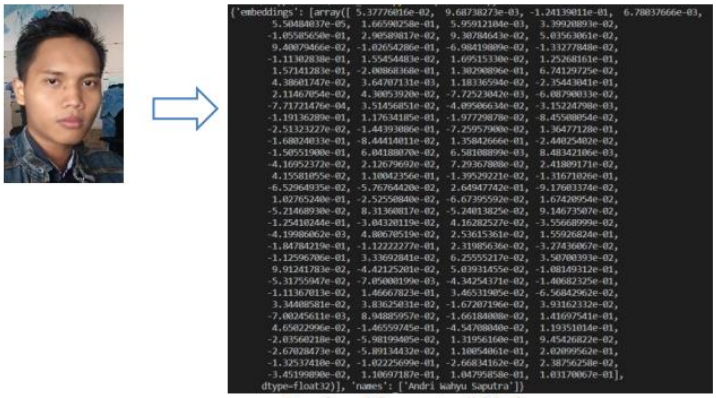
```
# compute the (x, y)-coordinates of the bounding box for the face
box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
(startX, startY, endX, endY) = box.astype("int")

# extract the face ROI
face = image[startY:endY, startX:endX]
(fH, fW) = face.shape[:2]
```

After the addition of feature extraction, it is continued by extracting the characteristic values of each face using the Convolutional Neural Network (CNN) method. This process is carried out to change the image dataset into a vector that describes the face stored in a file with the .pickle format (embedding.pickle). The process of changing the image dataset into a 128 d vector is carried out using the following source code.

Potongan Program

```
# Image to Vector 128 d
faceBlob = cv2.dnn.blobFromImage(face, 1.0 / 255, (96, 96), (0, 0, 0),
    swapRB=True, crop=False)
embedder.setInput(faceBlob)
vec = embedder.forward()
# Save The Facial Embeddings + Names To Disk
data = {"embeddings": knownEmbeddings, "names": knownNames}
f = open(args["embeddings"], "wb")
f.write(pickle.dumps(data))
```

Contoh ekstraksi 128 nilai ciri yang akan disimpan dalam *database* dapat dilihat pada gambar 4.2.



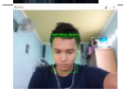
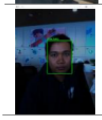
Gambar 4.2 : vector 128-d

The next process is the classification process. In the classification process, testing uses the results of the embedding process in the form of vector values from an image. Then the vector is forwarded through the SVM recognition model, the result of which is the prediction of the image. The next step is to take the highest probability value from the prediction results obtained. Then continue to the label encoder process to get the name of the image.

Face Detection Testing

This test is conducted to see how accurate the facial image detection is using the Caffe Deep Learning algorithm that will be used for the recognition process in recording the face of the application user. Table 2 shows the test results with several criteria.

Tabel 2. Pengujian Deteksi Wajah

Citra Uji	Jenis Pengujian	Hasil	Keterangan
	Citra wajah tampak depan	Terdeteksi	Ketika wajah dengan pose menghadap depan, sistem masih mendeteksi keberadaan wajah.
	Citra wajah dengan posisi kepala menunduk	Terdeteksi	Ketika wajah dengan pose menghadap depan dan kepala menunduk, sistem masih mendeteksi keberadaan wajah.
	Pencahayaan Normal	Terdeteksi	Ketika pencahayaan dari citra tersebut normal, sistem dapat mendeteksi keberadaan wajah.
	Pencahayaan Redup	Terdeteksi	Ketika pencahayaan dari citra tersebut redup, sistem masih dapat mendeteksi keberadaan wajah.
	Pencahayaan Gelap	Terdeteksi	Ketika pencahayaan dari citra tersebut gelap, sistem masih dapat mendeteksi keberadaan wajah.

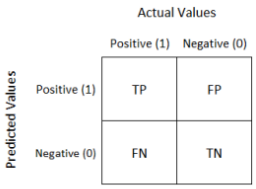
Facial Recognition Testing

In this study, there are 250 images with a composition of 150 facial images as training data and 100 images as test data as shown in Table 3.

Tabel 3. Dataset

Jumlah Sample Citra	Citra Per Orang	Total	
25	10	250	
Citra Latih	Citra Uji	Total Citra Latih	Total Citra Uji
6	4	150	100

This test is to measure the level of precision, recall and accuracy using the Confusion Matrix. In general, precision, recall and accuracy can be formulated based on the following image [13].



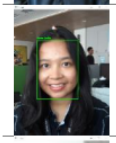
Gambar 6. Confussion Matrix

Precision, used to calculate the accuracy between the requested information and the answer given by the system. Recall, used to measure the level of success of the system in rediscovering information. Accuracy, used to compare the level of closeness between the predicted value and the actual value. Based on Figure 6, precision, recall and accuracy have their own calculation methods. The calculation of the value can be calculated using the following equation:

Precision = TP/(TP+FP) ×100% Recall = TP/(TP+FN) ×100% Accuracy = (TP+TN)/(TP+TN+FP+FN) ×100%

As with the facial detection testing process, the facial identification testing process is also carried out with various criteria as listed in Table 4.

Tabel 4. Pengujian Identifikasi Wajah

Citra Uji	Jenis Pengujian	Hasil	Keterangan
	Citra wajah tampak depan	Dapat Dikenali	Ketika wajah dengan pose menghadap depan, sistem dapat mengidentifikasi wajah.
	Citra wajah dengan posisi kepala menunduk	Dapat Dikenali	Ketika wajah dengan pose menghadap depan, sistem dapat mengidentifikasi wajah.
	Pencahayaan Normal	Dapat Dikenali	Ketika wajah dengan pose menghadap depan, sistem dapat mengidentifikasi wajah.
	Pencahayaan Redup	Dapat Dikenali	Ketika wajah dengan pose menghadap depan, sistem dapat mengidentifikasi wajah.
	Pencahayaan Gelap	Dapat Dikenali	Ketika wajah dengan pose menghadap depan, sistem dapat mengidentifikasi wajah.

From the results of testing the facial recognition system using the CNN method that has been carried out, the percentage of accuracy of the facial recognition system using test data is 98%. From the results of the test, there are 2 images that are not recognized by the system because the distance between the object and the camera is too close. This percentage of accuracy is calculated using the following equation:

Accuracy =$Jumlah data\ uji\ yang\ dikenali\ Jumlah data\ uji\ x$100%

Accuracy = 98 100x100% = 98%

Then the percentage results of the facial recognition system using the Confusion Matrix test produced a precision value of 98.4%, recall of 98% and accuracy of 99.84%.

To predict the precision, recall and accuracy values, they can be calculated from Table 4. With the first image test sample.

Tabel 4. Sample Hasil Percobaan Menggunakan Confusion Matrix

N = 100		Nilai Sebenarnya	
		TRUE	FALSE
Nilai Prediksi	TRUE	4	0
	FALSE	0	96
TOTAL		4	96

Dengan perhitungan sebagai berikut :

$$Precision = \frac{4}{4+0} \times 100\% = 100\%$$

$$Recall = \frac{4}{4+0} \times 100\% = 100\%$$

$$Accuracy = \frac{4+96}{4+96+0+0} \times 100\% = 100\%$$

Discussion

The CNN algorithm is one of the deep learning algorithms that focuses on studying the characteristics of an object in more depth. One of the most important parts of the success of this algorithm in classifying an object is the amount of data. The more data trained in the training process, the higher the accuracy results obtained. Likewise, this CNN algorithm is not optimal when the trained data is small. Therefore, additional extraction operations are needed to improve the results. In this study, there are 2 additional feature extraction processes, namely ROI and Bounding Box. These characteristics are used to reduce the high processing time. In carrying out the facial recognition process, the CNN algorithm managed to get an accuracy level of 98%. In this facial recognition system, the results obtained should still be able to be improved because the amount of data used in this study is too small.

CONCLUSION

Based on the results of the study that has carried out the design and planning of the Face Recognition program and through the tests that have been carried out, it can be concluded that Face Recognition technology can be done using Convolutional Neural Network (CNN). To improve CNN performance, the author uses several optimization techniques to train CNN. Pre Trained CNN with some additional data to improve the network's generalization ability, which requires less time to extract facial features from the target data set. The process of creating this application with the stages of creating Face Recognition, namely image acquisition, preprocessing, extraction, classification, and identification of image data. These stages are made with the python programming language. This study successfully used Face Recognition by 25 people with a facial dataset consisting of 10 facial data for each person. The dataset is separated into 2 stages of data, namely training data and test data. The results of implementing deep learning using the Caffe Deep Learning framework for the face detection process produced a percentage of 100% with predetermined criteria. Meanwhile, the Convolutional Neural Network method used

for the classification process produced a percentage of 98%, precision of 98.4%, recall of 98% and accuracy of 99.84%.

THANK-YOU NOTE

All parties who contributed to the research conducted are written in this section. And . Suggestions The following are suggestions that can be used as considerations for further research, namely to obtain better results can be done by adding different features or methods. With the development of mobile-based technology, it is necessary to try optimizing this facial recognition process on mobile devices. While in terms of hardware, it can be done by improving the quality of the camera used to obtain facial images and improving the quality of the computer used in the process of taking facial images.

BIBLIOGRAPHY

[1] A. Santoso and G. Ariyanto, “Implementation of Keras-Based Deep Learning for Face Recognition,” Emit. J. Tech. Electro, vol. 18, no. 01, pp. 15–21, 2018, doi: 10.23917/emitor.v18i01.6235.

[2] DLZ Astuti and Samsuryadi, “Study of Facial Expression Recognition using PCA and CNN Methods,” Annu. Res. Semin. Faculty of Comput. Science., vol. 4, no. 1, pp. 293–297, 2018.

[3] AP Putera and PN Primandari, “Design and Construction of Android-Based Online Attendance Application Using Deep Learning Method at PT. Pelabuhan Indonesia III (Persero),” 2020.

[4] M. Arsal, B. Agus Wardijono, and D. Anggraini, “Face Recognition for Bank Employee Access Using Deep Learning with CNN Method,” J. Nas. Technol. and Sist. Inf., vol. 6, no. 1, pp. 55–63, 2020, doi: 10.25077/teknosi.v6i1.2020.55-63.

[5] Y. Achmad, RC Wihandika, and C. Dewi, “Emotion classification based on facial features using convolutional neural network,” J. Pengemb. Technol. Inf. and Comput. Science., vol. 3, no. 11, pp. 10595–10604, 2019.

[6] S. Meenakshi, M. Siva Jothi, and D. Murugan, “Face recognition using deep neural networks across variations in pose and illumination,” Int. J. Recent Technol. Eng., vol. 8, no. 1 Special Issue4, pp. 289–292, 2019.

[7] MR Muliawan, B. Irawan, and Y. Brianorman, “Implementation of Face Recognition Using Eigenface Method in Attendance System,” J. Coding, Sist. Komput. Untan, vol. 03, no. 1, pp. 41–50, 2015.

[8] A. Kishore, “Designing Deep Learning Neural Networks using Caffe,” no. January 2015, pp. 1–17, 2015, doi: 10.6084/M9.FIGSHARE.1546481.

[9] J. Ali, “Design and Construction of Face Recognition System Using Principal Component Analysis Method,” vol. 1, no. 2, pp. 48–60, 2016.

[10] K. Santoso and GP Kusuma, "Face Recognition Using Modified OpenFace," Procedia Comput. Sci., vol. 135, pp. 510– 517, 2018, doi: 10.1016/j.procs.2018.08.203.

[11] M. Athoillah, “Face Recognition Using Multi Kernel SVM with Incremental Learning,” vol. 2, no. 2, pp. 84–91, 2017, doi: 10.15575/join.v2i2.109.

[12] K. Mujib, A. Hidayatno, and T. Prakoso, “Face Recognition Using Local Binary Pattern (Lbp) and Support Vector Machine (Svm),” Transient, vol. 7, no. 1, p. 123, 2018, doi: 10.14710/transient.7.1.123-130.

[13] MF Rahman, D. Alamsah, MI Darmawidjadja, and I. Nurma, “Classification for Diabetes Diagnosis Using Bayesian Regularization Neural Network (RBNN) Method,” J. Inform., vol. 11, no. 1, p. 36, 2017, doi: 10.26555/jifo.v11i1.a5452.