

Design and Development of Student Attention System using Face Recognition Based on MTCNN and FaceNet

Moch. Maulana A. C.¹, Hudiono², Yoyok Hero Prasetyo Isnomo³

^{1,2,3} Digital Telecommunication Network Study Program,

Department of Electrical Engineering, State Polytechnic of Malang, 65141, Indonesia

¹ andikapetek87@gmail.com ² hudiono@polinema.ac.id ³ yoyok.heru@polinema.ac.id

Abstract — Employees use their level of attendance or absence to demonstrate their presence at work or absence from it in an agency. This absence is connected to how discipline is applied, which is decided by each organization or institution. It can be inferred from this that student absenteeism in a setting where there is activity serves to increase discipline and demonstrate attendance. With IP Camera technology, it can be applied to the attendance system using the MTCNN method as face detection and FaceNet to extract high-quality features from the face. The system created can detect faces at a distance of 40 cm – 180 cm with an accuracy of 90.5% and can detect more than 1 object in 1 frame so that the IP Camera function is in accordance with the design in real-time. Tested the object of 3 pairs of twin faces produces a maximum accuracy of 90% where the level of match between the faces and the data is appropriate.

Keywords — Attendance system, FaceNet, IP camera, MTCNN

I. INTRODUCTION

The level of attendance or absenteeism is an activity or routine carried out by employees to prove that they are present or not present at work in an agency. This attendance relates to the application of discipline determined by each company or institution [1]. Thus, it can be concluded that absenteeism in an activity environment has the function of increasing discipline and showing student attendance.

The biometric system to identify a person with a body part that already exists in a person is an interesting thing to study [2]. This system identifies and recognizes a person by using one of the physical characteristics, namely the face. The system requires equipment for the identification process of a person by using an IP Camera that can take the shape of a person's face within a certain distance [3]. Face recognition has several advantages in identifying, it can be in the form of face images, photos, or image data taken directly through an IP Camera [4] at a certain distance.

Many methods can be used to recognize faces, one of which is by applying deep learning methods. In recent years, deep learning methods have won several contests in pattern recognition and machine learning [5]. One application of deep learning methods is artificial neural networks (Artificial Neural Networks) [6]. A standard neural network consists of many connected processors called simple neurons, each of which produces a sequence of real-valued activations [7].

Deep learning methods that can be used to recognize faces include the Convolutional Neural Network (CNN) method [8]. This CNN method [9] has several different types depending on the architecture formed, one of which is the Siamese Neural Network (FaceNet) [10]. Siamese Neural Network (FaceNet) has good potential to be used in facial recognition [11]. Because with this method, facial features can be extracted properly using two twins Convolutional Neural Networks, then

the distance from the two data extraction results from the two faces is calculated to get the similarity value [12].

In addition to the Siamese Neural Network (FaceNet) method, many other methods can be used to recognize faces, including the Direct Correlation Method, Eigenface, and Fisherface [13]. However, the three methods have fairly high error rates. Fisher's face has error rates of 20.1%, followed by Eigenface (25.5%) and Direct Correlation (25.1%) [14]. In addition, Direct Correlation also requires large processing and storage time [15]. Therefore, the authors chose to use the Siamese Neural Networks (FaceNet) method to solve the issue of repeatedly counting the same person throughout the face recognition process, the outcomes of facial recognition are then utilized to determine the number of people's faces. This is anticipated to enhance discipline and boost the efficiency of time in the tracking of students' attendance.

II. METHOD

A. Block Diagrams

Before IP cameras do data retrieval, the first thing the user does is process personal data registration and registration process face use the IP camera then the system will save on the system database. The stage dual system carries out the data training process with the use method. Later FaceNet system will carry out the process of mapping and introducing faces. Stage three users perform the attendance process on the face recognition feature then the system will compare with the results of training data. The process is described in Fig. 1.

In Fig. 1, the input made by the user is explained, there is a face scan process, then a face cropping process, then data training and face recognition. After that the application will process the data on the system. The resulting output is in the form of new data storage, face crop results, training with method FaceNet, and comparing training data results.

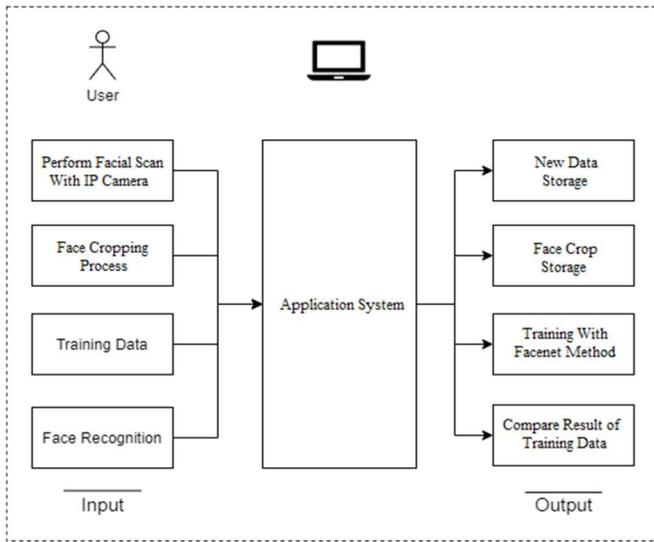


Figure 1. Block Diagram

B. Procedure MTCNN method

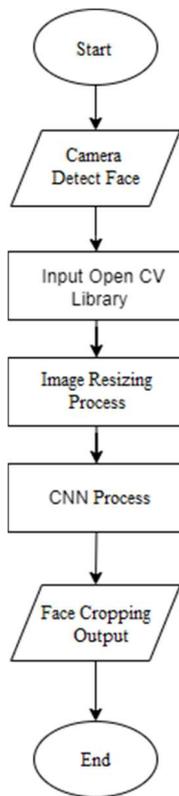


Figure 2. MTCNN Method Flowchart

Fig. 2 describes the process for the MTCNN method explained as follows. Taking eyes, nose, and mouth through a results detection camera on the face. Input the dedicated open CV library as a processing image. Next, the process of changing the picture's original face scan changed becomes size 12x12x3 as an input CNN process. In part, this system captures streaming results from an IP camera, then capture the image processed with a method where from results detection face

could obtain coordinate face on the picture as score x, y and score w, h where w is width and h are high, after that picture cropped by score x, y, w, h the picture already cropped then changed Becomes size 160x160 pixels. General parameters used in this process namely:

1. size (f) represents the height and width of the filter used. The filter size also represents the size of the kernel that will be used.
2. Stride(s) in size step to take when crossing horizontal and vertical lines.
3. Padding (p) is used to maintain the height and width of pictures because after processing size the input image is reduced.

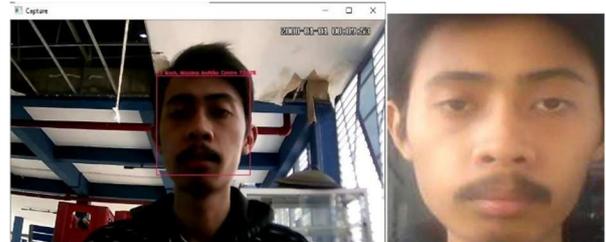


Figure 3. Cropping Process Using CNN

The result of the face cropping process using CNN with detect eyes and nose produces the crop face output described in Fig. 4.

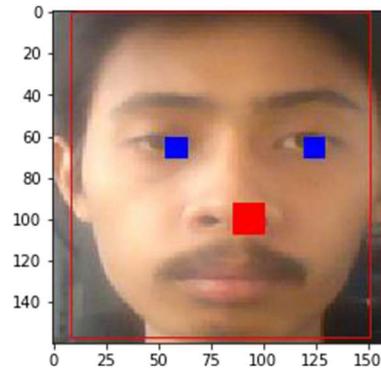


Figure 4. Face Cropping Results

Fig. 4 is inputted in the photo sample directory as the output of the MTCNN process in jpg format, looping the conducted as much as 50 times the results of the capture. Distinguishing Parameters face based on application script as follows:

1. Face Capture Parameters

```
138 video_capture = cv2.VideoCapture(0)
```

Figure 5. Video Capture

First parameter used for catching live stream with a camera. OpenCV provides a very interface simple this. With the application, uses the laptop's built-in IP Camera, changes it to grayscale video, and displays it. To capture video necessary, make an object Video Capture. An argument could index a device or video file name. The device index is the only number for determining the camera. Study this only using 0 (or 1) as well as one camera will be connected.

hundreds or thousands. So, the method *KNeighborsClassifier* is very suitable applied to the application.

C. Program Planning Method FaceNet

After knowing the methods and procedures performed. Next is the implementation method the use programming language *Python*. Following this is the code used procedure:

1. Add Student Data

```

detector = MTCNN()
video_capture = cv2.VideoCapture("rtsp://admin:admin@192.168.11.1:554/video1?channel=1&id=1&ids=1")
e count <= 50:try:
check, frame = video_capture.read() faces =
detector.detect_faces(frame) for result in
faces:
x, y, w, h = result['box']
face = frame[y-5:y+h+5, x-5:x+w+5] resized_face =
cv2.resize(face, (160,160))

cv2.imwrite(os.path.join(input_directory, name + str
(count) + '.jpg'),resized_face)
cv2.rectangle(frame, (x,y), (x+w, y+h),(0,0,255), 2)
count += 1

# show the output frame
res_size = ResizewithAspectRatio(frame, height=400)
cv2.imshow("Frame",res_size)
key = cv2.waitKey(1)if key ==
ord('q'):
break

pt Exception as e:pass
video_capture.release()
cv2.destroyAllWindows()
    
```

Figure 12. MTCNN Process

Fig. 12 about the coding of the process where *the system* will do recording live to take a picture to generate an image capture with a size of 160x160 pixels later picture will capture as much as 50x and be saved into training data.

2. Procedure Method FaceNet

The process of training data from Fig. 13 is explained as follows:

- a. Face Cropping

In this process, the system determines features of every face in pictures like eyes, nose, and mouth. Then the alignment process makes the slanted image will face straightforward.
- b. DNN inception resnet training V-1

In this process the system takes the output of the cropping file face that is a feature from the picture and will be studied by auto on layer *Deep CNN* (Inception Resnet V1) then the results learned features will be entered to layer *Fully Connected Layer* (FC). The FC layer is well trained so that produces good feature *vector* (*embedding*).

- c. L2 Normalization

In the normalization process used for the count the resulting distance goes more vector coordinates short then will continue with the embedding process.
- d. Embedding Process

The embedding process is vector that represents extracted features from face. This then could compare with generated vector for face another. For example, another vector close to (with size certain) may be the same person, whereas another distant vector (with size certain) may be different people.
- e. Triplet Loss Process

Process of training output from embedding that will be compared with inputs. Then will validate the face with training data.
- f. Output

Produce appropriate personal data identity with image input original.

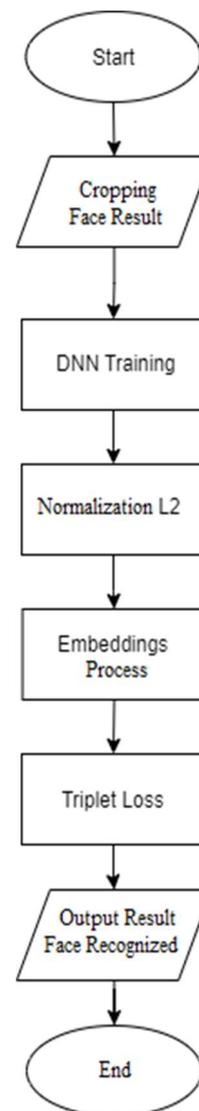


Figure 13. Flowchart FaceNet

III. RESULTS AND DISCUSSION

A. Result of Using a Mask

TABLE I
RESULT OF USING A MASK

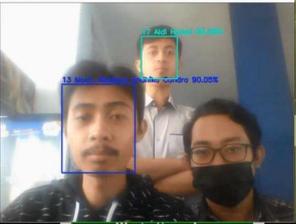
No	Picture	Information
1		System could recognize user moment wearing a mask
2		System no could recognize user moment wearing a mask
3		System only can recognize one masked user
4		System only can recognize users who don't wearing a mask

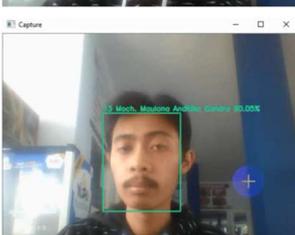
Table 1 describes the testing process moment use a mask. On testing this could see there are two users who wear masks 2 users who don't wear a mask. Then more systems sensitive against users who do not use a mask so that the data fits with that user. Result of testing wearing a mask is system can detect will, but system too sometimes no can detect face at the moment use of masks.

B. Result of Head Tilt

The test aims to be able to make it easier at the moment detective face. Following testing tilt head. Test on is results from testing slope head done at an angle to right, left, up, down, and straight to front. At the moment head directed to on the system can't detect then next at an angle to lower the system can't detect face then head directed to left neither system can detect face when head faced to the right the system doesn't could detect object last face facing to front system can recognize appropriate.

Results of the testing could conclude the system can't be used if face no facing to the camera so that when using the user application must focus on the front-facing camera so you can be detected and recognized his data.

TABLE II
RESULT OF HEAD TILT

No	Picture	Information
1		Not Detected
2		Not Detected
3		Not Detected
4		Not Detected
5		Detected

C. Result of Multiple Face Detection

Fig. 14 explains that the application could detect and recognize more than 1 object's face with good. With use, MTCNN and FaceNet methods system can detect in real-time and then will display name and accuracy data equation with

object. From testing, many faces produce a percentage of more than 90% based on the dataset.

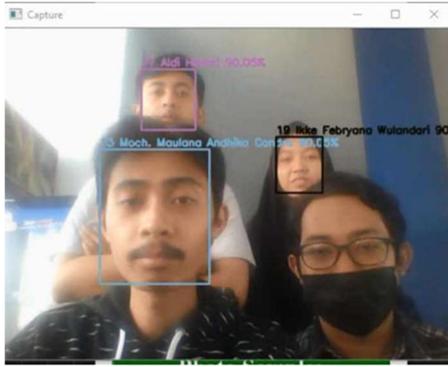


Figure 14. Multiple Face Detection

D. Result of Twins Object

TABLE III
RESULT OF TWINS OBJECT

No	Image 1	Image 2
1		
2		
3		

Testing the data Table 3 explains about results testing with object face twin, where the user will carry out the detection process later face, will match with the appropriate data. The system could detect face twins with produce a percentage maximum of 90% were level compatibility face with appropriate data. Data in column picture two has been identified by 90% with pictures before already stored in the system. Test object twin conducted 3 pairs of different twins. There is a 10% difference between each twin. FaceNet here plays a very role important in testing object twin because determining the score for measure level similarity face. Each neural network will use pictures as input then each existing layer will process differences or similarities from the Images.

E. Result of Accuracy

TABLE IV
RESULT OF ACCURACY

No	Picture	Accuracy
1		73.52%
2		89.37%
3		89.89%
4		89.93%
5		89.75%

Accuracy testing was conducted on 5 people where testing this conducted by objective. By doing a face scan will output name and accuracy of taking an image. Research has results where above 70% accuracy already in accordance so that application already worth used.

IV. CONCLUSION

Student Attendance System Using Face Recognition Based on MTCNN and FaceNet could run as designed and could be used without having to use an ID. The system created could detect faces at a distance of 40 cm – 180 cm with an accuracy of 90.5% and could detect more than 1 object in 1 frame so that the IP Camera function was in accordance with the design in real-time. Tested the object of 3 pairs of twin faces produces a maximum accuracy of 90% where the level of match between the faces and the data is appropriate. FaceNet played an important role in testing twin objects in order to determine the value to measure the level of facial similarity. In this study the FaceNet method ran according to the system so that the identification and classification of data is appropriated.

REFERENCES

- [1] H. Haryanti, S. Muchsin, and J. Administration Public, "Implementation of Improvement Models Discipline Employee Fingerprint Based On MTSN Nagekeo Regency Nagekeo Ntt (Study Case in MTs. N Nagekeo Regency Nagekeo NTT)", *J. Respon Public*, vol. 13, no. 3, pp. 78–84, 2019.
- [2] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," *Neural Networks*, vol. 61, pp. 85–117, Apr. 2017, doi : 10.1016/j.neunet.2014.09.003.
- [3] Y. Arkhiansyah and D. Setiawan, "Realization cctv Intelligent Based on Microcontroller and Real Time 3D Face Recognition," *J. Inform.*, vol. 15, no. 2, pp. 188–197, 2019.
- [4] KS Wibawa, T. Information, F. Teknik, U. Udayana, and B. Jimbaran, "Visual Security Monitoring and Control System", pp. 1–8, 2017.
- [5] "Introduction Face Using the Facenet Model for Presence Employees", ITS Repository. <https://repository.its.ac.id/76660/> (accessed Jul. 13, 2021).
- [6] K.P. Danukusumo, "Deep Learning Implementation Using Convolutional Neural Networks for Classification of Temple Image Based on GPU ", Jul. 2017.
- [7] S. Sena, "Introduction to Deep Learning Part 7: Convolutional Neural Network (CNN)," 2017. <https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94> (accessed Dec. 14, 2018).
- [8] "Understanding of Convolutional Neural Network (CNN) — Deep Learning | by Prabhu | Medium." <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148> (accessed Jul. 13, 2021).
- [9] "Rectified Linear Units (ReLU) in Deep Learning | Kaggle." <https://www.kaggle.com/dansbecker/rectified-linear-units-relu-in-deep-learning> (accessed Jul. 13, 2021).
- [10] "Viewing Activation Function. tie it up Knowledge with Write it down | by Pramesti Hatta K. | Medium." <https://medium.com/@opam22/menilik-activation-functions-7710177a54c9> (accessed Jul. 13, 2021).
- [11] "Impact Smartphone Technology Against Behavior of Parents in the Village Toure Subdistrict Tompasso, Moku Acta Diurna Communication ." <https://ejournal.unsrat.ac.id/index.php/actadiurnakomunikasi/article/view/10929> (accessed Jul. 13, 2021).
- [12] A. Vedaldi and K. Lenc, "MatConvNet - Convolutional Neural Networks for MATLAB," Dec. 2014, Accessed: Jul. 13, 2021. [Online]. Available: <http://arxiv.org/abs/1412.4564>.
- [13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 56, pp. 1929–1958, 2014, Accessed: Jul. 13, 2021. [Online]. Available: <http://jmlr.org/papers/v15/srivastava14a.html>.
- [14] Z. Zhang, "Derivation of backpropagation in convolutional neural network (CNN)", Univ. Tennessee, Knoxville, TN, Oct. 2016, Accessed: Jul. 13, 2021. [Online]. Available: Derivation of backpropagation in convolutional neural network (CNN).
- [15] Han, "Chapter II Foundation Theory", *J. Chem. inf. Models*, vol. 53, no. 9, pp. 1689–1699, 2019.