



## Face Recognition of Indonesia's Top Government Officials Using Deep Convolutional Neural Network

Umar Aditiawarman<sup>1</sup>, Dimas Erlangga<sup>2</sup>, Teddy Mantoro<sup>3</sup>, Lutfil Khakim<sup>4</sup>  
<sup>1,2,4</sup> School of Computer Science, Nusa Putra University  
<sup>3</sup> Faculty Science and Technology, Sampoerna University  
<sup>1</sup>umar.aditiawarman@nusaputra.ac.id, <sup>2</sup>dimas.erlangga@nusaputra.ac.id

### Abstract

*Facial recognition is a part of Computer Vision that is used to get facial coordinates from an image. Many algorithms have been developed to support Facial Detection such as Cascade Face Detection using Haar-Like features and AdaBoost to classify its Cascade and Convolutional Neural Network (CNN). Face recognition in this study uses the Deep Convolutional Neural Network (DCNN) method, and the output of this method is the measurement value of the face. In the model training process, Triplet Loss from Triplet Network Deep Metric Learning is used to get good face grouping results. The value of this face measurement will then be measured using the Euclidean distance calculation to determine the similarity of the input face from the dataset. This Research is using 6 images of Government officers in Indonesia to determine the accuracy of the model when there is a new picture of these officers inputted into the training machine. The result provides a 94% accuracy level with a variety of face positions and levels of brightness.*

*Keywords: deep convolutional neural network; face recognition; facenet; haar cascade classifier*

### 1. Introduction

Computer vision is a field of Artificial Intelligence (AI) that enables computers to gain information from digital images and videos. This section of Computer Science is based on how human sees an object and translate the object in order to identify by its characteristics [1]. Face Recognition is a part of Computer Vision that is used to get facial coordinates from an image. Many algorithms have been developed to support facial detection such as Cascade Face Detection which was developed by Viola and Jones [2] using Haar-Like features and AdaBoost to classify its Cascade, which has better performance when it used in real-time situations.

There are several concerns regarding the effectiveness of face recognition when it deals with different angles or poses in images. However, several studies [3]- [4] show that Cascade methods and Deformable Part Models (DPM) can minimize the false rejection in a variety of faces used in Face Recognition. Consequently, it requires high-level computation and annotation in the training stage.

Later, the Convolutional Neural Network (CNN) appeared and had outstanding progress in computer vision, inspired by deep learning methods of face

detection. Several past studies have been conducted using CNN for face detection [5], [6] until a Multi-Task Convolutional Neural Network was proposed with facial landmarks and enhanced by using techniques that have been done by several previous researchers, thus producing coordinates on faces that fit even on a small size in the image [7].

Face detection is performed to get the coordinates of the human face, then facial recognition is performed to find out the identity of the human face by comparing it with their facial features. Several known face recognition algorithms such as the Local Binary Pattern Histogram (LBPH) algorithm [8], Fisherface, SIFT and SURF, and now Deep Learning techniques offer good accuracy., The development of facial recognition algorithms aims to increase the accuracy of face identification in the image.

Face recognition in this application will use the Deep Convolutional Neural Network (DCNN) method and the output of this method is the measurement value of the face. In the model training process, Triplet Loss from Triplet Network Deep Metric Learning is used to get good face grouping results [9] The value of this face measurement will then be measured using the Euclidean

distance calculation to determine the similarity of the input face from the dataset [10].

This facial recognition model is trained by FaceNet with the euclidean embedding learning method using the DCNN. This model is trained using Stochastic Gradient Descent with standard backprop and AdaGrad. This model uses rectified linear units as a non-linear activation function and uses two architectures, namely the convolutional layers architecture of Zeiler & Fergus and GoogLeNet style Inception models.

This Research is using 6 images of Government officers in Indonesia to determine the accuracy of the model when there is a new picture of these officers inputted in the training machine. These government officers were chosen randomly representing their position from the president, coordinating ministry, ministry, and governor.

In this area of Face Recognition using Deep Convolutional Networks, many relevant researchers have been developed in recent years. Panji Purwanto, Burhanudin Dirgantoro, and Agung Nugroho Jati in 2015 [11] proposed a face detection and recognition application on a surveillance camera as a hazard detector. The face detection method used in this application is the Haar Cascade Classifier Like Feature proposed by Viola and Jones.

The facial recognition method used by the author is Fisherface. There are three steps after face detection in this method, including Principal Component Analysis (PCA), then Fisher Linear Discriminant (FLD) and classification.

The PCA and FLD calculation modules are used to form Fisherface sets which later gave different weights to the faces. After getting the face weights, the input faces will be compared with the face weights in the dataset using the Euclidean distance calculation. They have tested 66 input images providing an accuracy of 81.82%.

Wibowo Joko Nuryanto in 2017 [12] made a facial recognition application using the Speeded Up Robust Features method. The face detection process started from determining the parameters, processing the Haar Detect Object feature, then converting the image into a gray image and cropping the image. The dataset used has a total of 30 images per person and tested on 50 images with different levels of height, distance, and lighting. According to the table match the detection and recognition statistics. The highest equation value (1.0) is found in the same image, and when different images with the same face owner have a value of 0.4, some even have a value of only 0.1. From the displayed values, the accuracy of facial recognition decreases drastically when the images obtained are different.

Another research on facial recognition application was conducted by Sinar Monika, Adul Rakhman, and

Lindawati [13]. These researchers proposed a facial recognition application for home security in real-time using the Principal Component Analysis / Eigenface method. Researchers did not include face detection methods in the paper, and used the Eigenfaces method for face recognition. Researchers have problems with the lighting factor, where this lighting can be a differentiating factor in Eigenface. Some other factors are blur, stretch, changes in facial expressions, and shooting from different angles. The database training process uses 30 face images per person from a total of 6 samples. Tests were carried out and obtained 88% accuracy results. Several factors that become the main problem in face detection using this eigenface method include lighting, angle of taking facial images, similarity of eigenface values between faces, and facial expressions.

Subsequent research was carried out by Sayeed Al-Aidid, Daniel S. Pamungkas [14]. The researchers used Haar Cascade algorithm and the Local Binary Pattern Histogram (LBPH) for facial recognition. The face detection method used is the Haar Cascade Classifier Like Feature proposed by Viola and Jones. The LBPH value is compared with the dataset using the Euclidean distance calculation. In the dataset creation process, each person's face will be filled with at least 20 face images with various poses and different angles. Then the face dataset will be converted into a grayscale image, after that the image of the face will be extracted to get the histogram value in the form of an array for each person. Here the author does not include the results of the analysis of the accuracy of facial recognition, the author only compares the detection of human faces and non-human faces. Human faces are not detected at a distance of 160 cm.

Adeshina SO *et all* [15] proposed face detectors using Haar-like and LBP features. MATLAB's `trainCascadeObjectDetector` function is used to train. 2577 positive face samples and 37,206 negative samples for a range of False Alarm Rate (FAR) values (i.e., 0.01, 0.05, and 0.1). The findings show that However, the study shows that the Haar cascade face detector is the most efficient (100% True Positive Rate (TPR) face detection accuracy) when tested on a set of classroom images dataset. While tested using deep learning ResNet101 and ResNet50, it outperformed the average performance of Haar cascade by 9.09% and 0.76% based on TPR, respectively. The TPR of the proposed algorithm is 92.71% when tested on images in the synthetic Labeled Faces in the Wild (LFW) dataset and 98.55%.

Sanjudharan *et all* [16] proposes a facial recognition method to improve home security by using an algorithm for face detection and recognition (Haar Cascade Classifier). Meanwhile, Anirudha [17] aspires to present the comparison of two face recognition

techniques Haar Cascade and Local Binary Pattern edified for the classification. As a result the accuracy of Haar Cascade is more than the Local Binary Pattern but the execution time in Haar Cascade is more than Local Binary Pattern. On the other hand, Tejas *et al* [18] build their own Custom HAAR-Cascade Classifier using "Cascade Trainer GUI (a tool designed by Amin Ahmadi) to detect face/faces in any given image/images and also create a dataset which include positive and negative samples to use during training purpose. They are also demonstrating how to retrain the classifier after analyzing the error matrix after each detection stage and how to increase the accuracy of the classifier in detection work.

Lia Farokhah [19] compared three methods in face detection, namely OpenCV Haar Cascade, OpenCV Single Shot Multibox Detector (SSD) and Dlib CNN. Face detection is focused on five challenging conditions, namely face detection in head position obstacles, wearing face masks, lighting, background images that have a lot of noise, differences in expression. Data testing is taken randomly on google with reference to one image consisting of more than one detected face with wild condition. The results of the comparative analysis in wild condition show that the OpenCV haar cascade has more weaknesses with a performance percentage of 20% compared OpenCV SSD and Dlib CNN method. Performance results of SSD and Dlib CNN have the same performance in the five conditions tested, which is about 80%.

Abd El-Hafeez *et al* [20] concerned with face recognition in a video stream using Local Binary Pattern histogram with processed data. The system will detect faces by using a combination of Haar cascade files that uses skin detection, eye detection and nose detection as input of LBP to increase the accuracy of the proposed recognition system. Also, their system can be used to build a dataset of faces and names to be used in a recognition step. The experimental results have shown that the proposed system can achieve accuracy of recognition up to 96.5% which was better than the relevant methods. Then M. Sitorus and Nurul Fadillah [21] using Haar Cascade Classifier to detect many faces on real time situations.

## 2. Research Methods

There are two stages in this research to recognize the face. First, the dataset needs to be prepared and labeled using Haar Cascaded Classifier to identify images of the same patterns. This process should be performed before the FaceNet pre-processing to produce a signature from the targeted images. Second, the FaceNet able to read images during the train process with various dimensions. After the dataset is created it can be loaded into the network training. Once the training model

provide a good accuracy then it can be used for new data that can distinguish the faces from different sides.

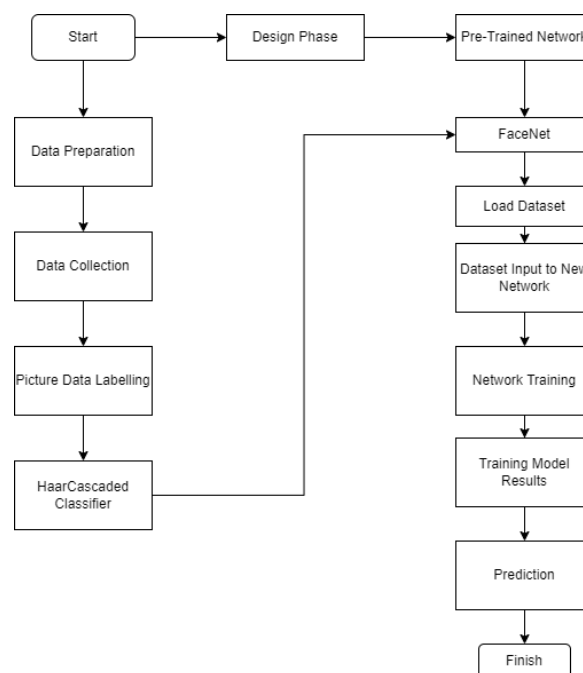


Figure 1. Research Method

### 3.1 Facenet Model Structure



Figure 2. Model Structure

Facenet network consists of a batch input layer and a deep CNN followed by L2 normalization, which results in the face embedding (Figure 2). This is followed by the triplet loss during training.



Figure 3. Triplet Loss Model

As can be seen in Figure 3, the Triplet Loss minimizes the distance between an anchor and a positive, both of which have the same identity, and maximizes the distance between the anchor and a negative of a different identity. Triplet loss formula can be defined in equation 1.

$$\sum_i^n [\|f(x_i^a) - (x_i^p)\|_2^2 - f(x_i^a) - (x_i^n)\|_2^2 + \alpha] \quad (1)$$

$x_i$  represent an image,  $f(x_i)$  represent the embedding of the image,  $\alpha$  represent the margin between positive and negative pairs.

### 3.2 The Dataset

This research uses six pictures of top government officer in Indonesia as seen in Figure 4, such as Joko Widodo, Airlangga Hartanto, Hadi Tjahjanto, Luhut Binsar Panjaitan, Anies Baswedan and Nadiem Makarim. These pictures are labeled with the name of the officials and classify using Haar Cascade Classifier.



Figure 4. Photo Dataset

### 3.3 Pre-Trained Network

The input parameter in the FaceNet pre-process is a photo input size of 160 X 160 pixels which is multiplied by 3 (three) or RGB. When providing input to FaceNet, FaceNet will train so that it will produce an output of 128 nodes or 128 numbers. These 128 numbers characterize the face that has been input. If the input face is different then the 128 X 1 matrix will be different, if the input face of the same person even though the pose is different, the resulting number is similar to the same number so it is called a "signature" or fingerprint.

Haar Cascade Classifier serves to detect how many faces are detected, Haar Cascade has certain patterns called features [10]. The features that will be tested on existing images, and from the suitability of the image with existing features and from the suitability of the image with the existing features, Haar Cascade will

know which ones are facial photos and which are not. In order for FaceNet to be able to read images during the train process, it is necessary to create a signature by changing the image size into a dataset from 640 X 480 pixels to 160 X 160 pixels, then changing the image from 3 dimensions to 4 dimensions so that it can be read when running the model. When entering the dataset into Haar Cascade it will generate input in the form of face detection and after processing it on Haar Cascade, FaceNet will generate a signature.

### 3.4 Load Dataset

The next stage is to load the previously created dataset, the function of the data load is so that the data that has been created can be processed to produce a signature. The result of the dataset load is, trainX is equal to 150 X 128 and trainY is 150 X 1. 150 is the entire image that has been processed, while 128 is the output number from FaceNet.

### 3.5 Input Dataset to Network with New Layer

The next step is to input the dataset into the network that has been created with several layers. The first layer is 128 because previously it was 128 numbers using Rectified Linear Unit (ReLU) activation and consisted of 20 nodes, the 2nd and third layers consisted of 10 nodes and ReLU activation. The last layer consists of 3 nodes and the activation is softmax.

### 3.6 Training and Model Results

After getting the dataset and creating a layer, the next step is to train the network that has been created so that it can distinguish faces from many sides. The epochs are 100 while the batch size is 150, because it consists of 150 images per input.

All the experiments above were trained using google colab so that the training process can run faster. The accuracy obtained from the training results is 94%. The training model is a model that will be used to detect images. This model has the extension (.h5) where in this model there are weights, labels of all classes and the deep learning architecture of the FaceNet model as a whole.

## 3. Results and Discussions

The facial verification model was trained using Google Colab. The evaluation of the given model is to try different values on the epoch, optimizer, and dense layer. The epochs were tested from 10, 20, 30, and 50, and the dense layers from 0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024. From these experiments and get good accuracy lies in 50 epochs and 512 dense layers provided 94% accuracy.

The optimizers that were tested were Adam and SGD, and the ones used were Adam Optimizer. As can be

seen in Figure 5. the epoch 30 and 50 do not have significant difference.

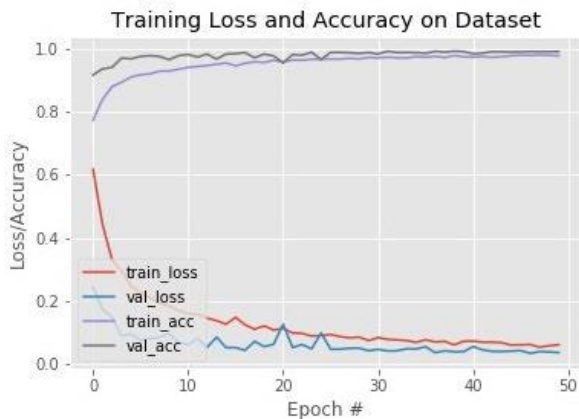


Figure 5. Training Results on verification models

The process of implementing the model into the application is by calling the model with the load\_model function. Here the existing models are the FaceNet model. The FaceNet model will later be used to calculate the face measurement value on the frame in real time to calculate its Euclidean value with the model in the classifier. After that, the face can be predicted by using the predict function with the model that has been made (Figure 6).

```
def findFaces(data):
    gbr1 = js_to_image(data)
    gbr = cv2.cvtColor(gbr1, cv2.COLOR_BGR2RGB)
    gbr = img_fromarray(gbr) # konversi dari OpenCV ke PIL
    gbr_array = asarray(gbr)

    wajah = HaarCascade.detectMultiScale(gbr1,1,1,4)

    for (x1,y1,w,h) in wajah:
        x1, y1 = abs(x1), abs(y1)
        x2, y2 = x1 + w, y1 + h

        face = gbr_array[y1:y2, x1:x2]

        face = img_fromarray(face)
        face = face.resize((100,100))
        face = asarray(face)

        face = face.astype('float32')
        mean, std = face.mean(), face.std()
        face = (face - mean) / std

        face = expand_dims(face, axis=0)
        signature = MyFaceNet.predict(face)

        min_dist=100
        identity=""
        for key, value in database.items():
            dist = np.linalg.norm(value-signature)
            if dist < min_dist:
                min_dist = dist
                identity = key

        cv2.putText(gbr1,identity, (x1,y1),cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 0), 1, cv2.LINE_AA)
        cv2.rectangle(gbr1,(x1,y2),(x2,y2), (0,255,0), 2)

    filename="photo.jpg"
    cv2.imwrite(filename, gbr1)

    return filename
```

Figure 6. Predict Face Code using Facenet

Next phase is data testing, are shown in Figure 7, 8, 9, 10, 11, 12 where in this research the model that has been trained using the dataset. In this phase, webcam is engineered to capture picture of the government officer then processed to the model to determine the identity of the person captured in the image. Captured image are stored in the system and processed as a new file including Recognition results of the person based on trained picture.

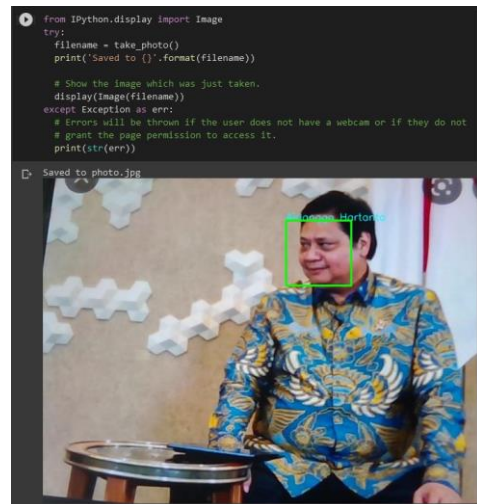


Figure 7. Data Testing on Airlangga Hartanto's Picture

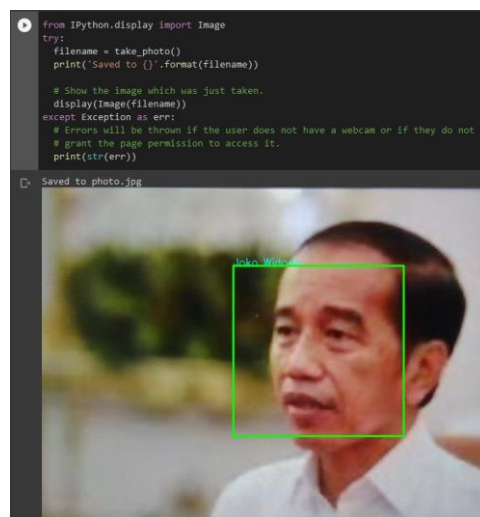


Figure 8. Data Testing on Joko Widodo's Picture

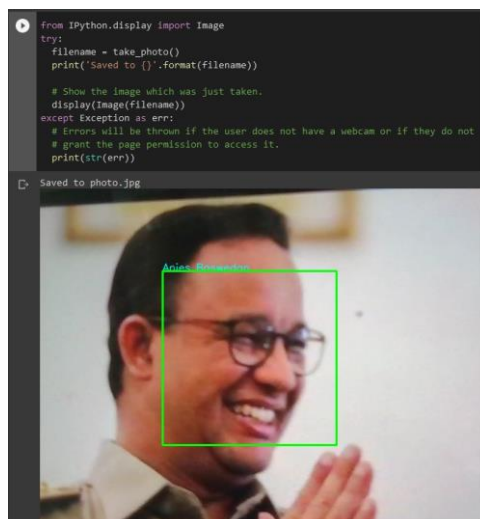


Figure 9. Data Testing on Anies Baswedan's Picture

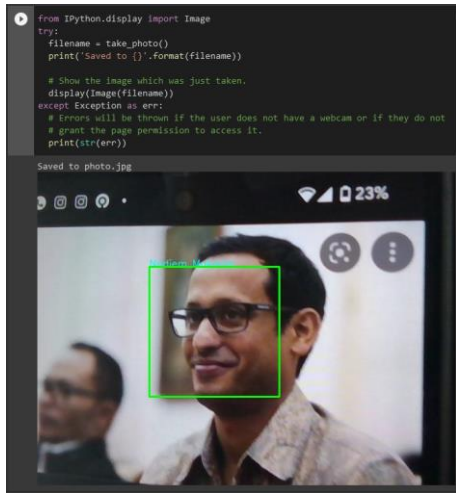


Figure 10. Data Testing on Nadiem Makarim's Picture

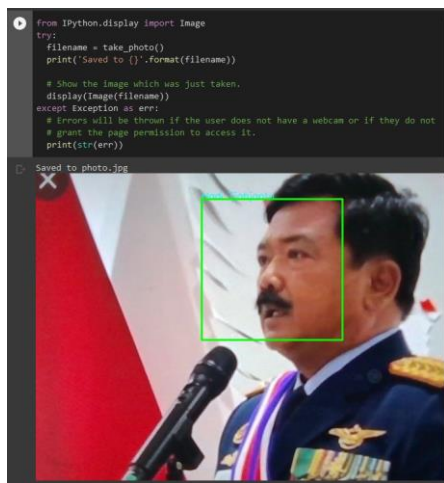


Figure 11. Data Testing on Hadi Tjahjanto's Picture

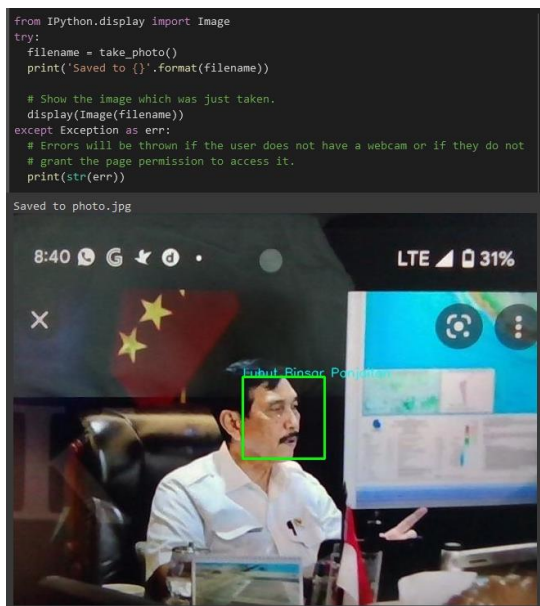


Figure 12. Data Testing on Luhut Binsar Panjaitan's Picture

## 4. Conclusion

Based on the analysis and results of data testing, FaceNet with Deep Convolutional Neural Network using Haar Cascade Classifier can determine faces of Government Officials in Indonesia with its effectiveness in face recognition using variety of face position and brightness of the government officer picture taken with accuracy levels of 94%. Theoretically, this study has enriched several findings in the field of face recognition using Haar Cascade Classifier algorithm. Practically, the result of this study has contributed to the mechanism of accurately recognizing government officials on the internet applications such as web and social media. Future research may use other algorithm to gain more accuracy level and add more datasets with similar kinds of faces to test the accuracy of Face Recognition and also create recognition embedded in Video mode in order to identify faces in real-time.

## Reference

- [1] W. S. Pambudi, B. Maria and N. Simorangkir, "Facetracker Menggunakan Metode Haar Like Feature," vol. 2, no. 2, p. 142–154, 2012.
- [2] B. Yang, J. Yan, Z. Lei and S. Z. Li, "Aggregate channel features for multi-view face detection," *IJCB*, 2014.
- [3] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," p. 1193–1197, 2001.
- [4] Q. Zhu, S. Avidan, M. C. Yeh and K. T. Cheng, "Fast human detection using a cascade of histograms of oriented gradients," *IEEE*, vol. 2, p. 1491–1498, 2006.
- [5] J. Yan, Z. Lei, L. Wen and S. Z. Li, "The fastest deformable part model for object detection," *IEEE*, pp. 2497–2504, 2014.
- [6] S. Yang, P. Luo, C. C. Loy and X. Tang, "From facial parts responses to face detection: A deep learning approach," *IEEE*, no. 3, p. 3676–3684, 2015.
- [7] K. Zhang, Z. Zhang, Z. Li, S. Member and Y. Qiao, "Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks Kaipeng," pp. 1–5.
- [8] N. M. Sharef, H. M. Zin and S. Nadali, "Overview and Future Opportunities of Sentiment Analysis Approaches for Big Data," *Journal of Computer Science*, vol. 12, no. 3, pp. 153–168, 2016.
- [9] S. Gao, H. Jinxing and Y. Fu, "The Application and Comparison of Web Services for Sentiment Analysis in Tourism," *12th International Conference on Service Systems and Service Management (ICSSSM)*, vol. 12, pp. 1–6, 2015.
- [10] A. Rahim, N. Hossain, T. Wahid and S. Azam, "Face Recognition using Local Binary Patterns (LBP)," *Glob. J. Comput. Science Technol. Graph. Vis.*, vol. 13, no. 4, pp. 469–481, 2013.
- [11] P. Purwanto, B. Dirgantoro and A. N. Jati, "Implementasi Face Identification Dan Face Recognition Pada Kamera Pengawas Sebagai Pendeteksi Bahaya," *eProceedings*, vol. 2, no. 1, p. 718–724, 2015.
- [12] W. J. Nuryanto, "Pengenalan Wajah (Face Recognition) Dengan Menggunakan Metode SURF (Speeded Up Robust Features)," p. 200, 2017.
- [13] S. Monika, A. Rakhman and L. , "Pengaman Rumah dengan Sistem Face Recognition secara Real Time Menggunakan

- Metode Principal Component Analysis," *SNATIF*, vol. 4, p. 35–40, 2017.
- [14] S. Al-Aidid and D. Pamungkas, "Sistem Pengenalan Wajah dengan Algoritma Haar Cascade dan Local Binary Pattern Histogram," *J. Rekayasa Elektr.*, vol. 14, no. 1, pp. 62-67, 2018.
- [15] A. SO, I. H., T. SS and H. SC, "Custom Face Classification Model for Classroom Using Haar-Like and LBP Features with Their Performance Comparisons," *Electronics*, p. 102, 2021.
- [16] S. M. M. S. and R. Priyansh, "Face Recognition System Based on Haar Cascade Classifier," vol. 29, pp. 3799-3805, 2020.
- [17] A. B. Shetty, D. Bhoomika and R. Jeevan Rebeiro, "Facial recognition using Haar cascade and LBP classifiers," *Global Transitions Proceedings*, vol. 2, no. 2, 2021.
- [18] T. Phase, "Building Custom HAAR-Cascade Classifier for face Detection," *International Journal of Engineering and Technical Research*, vol. 8, 2020.
- [19] L. Farokhah, "Perbandingan Metode Deteksi Wajah Menggunakan OpenCV Haar Cascade, OpenCV Single Shot Multibox Detector (SSD) dan DLib CNN," *RESTI*, vol. 5, no. 3, pp. 609-614, 2021.
- [20] A. El-Hafeez, T. & A. and Y. Mohany, "An Accurate System for Face Detection and Recognition," *JAMCS*, vol. 33, pp. 1-19, 2019.
- [21] M. Sitorus and N. Fadillah, "Multi Face Detection System Using Haar Cascade Classifier Method," *JICOM*, vol. 1, no. 1, pp. 1-5, 2020.