Real-Time Face Recognition Civil Servant Presence System Using DNN Algorithm

Yogi Angga Putra[1], Imelda Imelda*[2]
[1,2]Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur, Jakarta, Indonesia
e-mail: [1]1811500360@student.budiluhur.ac.id, **[*2]imelda@budiluhur.ac.id**

Abstrak

Pengenalan wajah telah menjadi topik yang berkembang di kalangan peneliti Computer Vision karena dapat memecahkan masalah kehidupan nyata, termasuk selama pandemi COVID-19. Pandemi menjadi alasan pemerintah Indonesia memberlakukan pembatasan sosial dan kontak fisik di tempat umum. Sebelum pandemi, sebagian besar sistem presensi berbasis sentuhan menggunakan sidik jari atau kartu Radio Frequency Identification (RFID). Solusi yang diusulkan dalam penelitian ini adalah mengidentifikasi real-time pengenalan wajah sistem presensi Pegawai Negeri Sipil menggunakan Deep Neural Network. Tujuannya untuk meminimalkan kontak fisik. Tahapan penelitian meliputi pengumpulan data, augmentasi dan preprocessing, pemodelan dan pelatihan CNN, evaluasi model, convert to OpenCV DNN, implementasi transfer learning, dan identifikasi data uji. Kontribusi penelitian ini melakukan pengujian variasi jarak dan posisi sehingga dapat mengenali wajah seseorang bahkan saat menggunakan masker dan kacamata. Model DNN ini menghasilkan nilai akurasi validasi sebesar 99,48% dan kehilangan validasi sebesar 0,0273 dengan proses pelatihan data sebanyak 10 kali. Pengujian variasi jarak, posisi, penggunaan masker dan kacamata pada pendeteksian MTCNN memberikan akurasi rata-rata untuk setiap pengujian berturut-turut sebesar 100%, 96%, dan 100%. Sebaliknya, akurasi rata-rata pengujian deteksi Haar Cascades secara berurutan adalah 100%, 85%, dan 100%.

Kata kunci—*Pengenalan Wajah, Deep Neural Network, real-time, MTCNN, Haar Cascades*

Abstract

Facial recognition has become a growing topic among Computer Vision researchers because it can solve real-life problems, including during the COVID-19 pandemic. The pandemic is why the Indonesian government has imposed social restrictions and physical contact in public places. Before the pandemic, most touch-based attendance systems used fingerprints or Radio Frequency Identification (RFID) cards. The solution proposed in this study is to identify real-time facial recognition of the Civil Service presence system using a Deep Neural Network. The goal is to minimize physical contact. The research stages include data collection, augmentation and preprocessing, CNN modeling and training, model evaluation, converting to OpenCV DNN, implementation of transfer learning, and identification of test data. This research contributes to testing variations in distance and position so it can recognize a person's face even when wearing a mask and glasses. This DNN model produces a validation accuracy value of 99.48% and a validation loss of 0.0273 with a data training process of 10 times. Tests for variations in distance, position, use of masks, and glasses on MTCNN detection provide an average accuracy for each trial of 100%, 96%, and 100%, respectively. Therefore, the average accuracy of the Haar Cascades detection test is 100%, 85%, and 100%.

Keywords— *Face Recognition, Deep Neural Network, real-time, MTCNN, Haar Cascades*

1. INTRODUCTION

Computer Vision is a part of *Computer Intelligence* that instructs data processing machines to study and interpret the visual world [1] . Facial recognition has become one of the most popular and growing topics in *Computer Vision* over the last few decades. With the help of video feeds or camera feeds from multiple capture devices, the computer can precisely detect, examine, identify, and classify objects in response to what it sees. Computer vision can now be applied to solve real-life problems in uncontrolled environments [2]. The ongoing COVID-19 pandemic is one example of an urgent situation that can be tackled using computer vision. This pandemic is an unprecedented crisis that has affected more than 228 countries worldwide by infecting more than 555 million people, and the number is still growing [3].

Corona Virus Disease 2019, commonly abbreviated as COVID-19, is an infectious disease caused by SARS-CoV-2, a type of *coronavirus*. The COVID-19 virus was first discovered in Wuhan, China, in December 2019 [4] . The Indonesian government implemented a new policy, the *leveling* Enforcement of Restrictions on Community Activities to suppress the spread of this virus [5]. The goal is to minimize physical contact in public places. People must also change some habbits, one of which is not touching objects that many commonly used.

Most attendance systems before the pandemic were touch-based, using fingerprints or RFID cards [6]. However, all these methods have several limitations. Therefore, the most suitable method to ensure full security and to save history records is through a smart face recognition system [7]. In addition, there is the application of a non-touch-based presence system using a *captive portal* or an application on a smartphone. Still, the absent person does not have a special/unique verification that it is accurate and can commit fraud in the form of an absentee. Therefore, it is necessary to overcome the shortcomings of the present system, especially during this pandemic, by using a facial recognition system to avoid physical contact and also attendance fraud because only the person concerned can only make attendance.

The solution that has existed in previous research [8] performs an analysis of facial readings on images and videos in *real-time* method *nave Bayes* Achieves 91.7% accuracy in recognizing images and 86.7% in real-time video. Using The DNN method instead of CovNets and use of haar cascade for extracting facial features resulted in an accuracy is not compromised in the proposed method as average accuracy obtained is 97.05% [9]. Utilize transfer learning by comparing three pre-trained convolutional neural networks SqueezeNet, GoogleNet, and AlexNet where they achieved a validation accuracy of 98.33%, 93.33%, and 100% respectively [10]. Many studies also use the CNN model to recognize face mask [11]–[17]. The other studies use the CNN method and the YoloV3 detector to detect and recognize faces with and without masks. The test compares the ResNet152V2, InceptionV3, Xception models with accuracy values, respectively: 99%, 97% and 99% [12]. There is another study whose video data comes from CCTV using motion detection and facial recognition. His research detects motion using the Accumulative Difference Images (ADI) method, detects faces using the Haar Cascade Classifier, features extraction and reduction using Speeded-Up Robust Features (SURF) and Principal Component Analysis (PCA) algorithms, and identifies faces using the Counter-Propagation Network algorithm (CPN). The success rate of motion detection is 92.6%, face detection is 76%, and face identification is 60% [18]. Research that uses image data from CCTV is also carried out for face classification by comparing the datasets of splitting Random Subsampling Validation (RSV), k-fold Cross Validation (k-FCV), Bootstrap Validation (BV) and Moralis Lima Martin Validation (MLMV) datasets. That research uses Convolutional Neural Network (CNN) and Support Vector Machine (SVM) algorithms. The comparison of datasets shows that the MLMV technique has the lowest performance compared to other techniques [19].

In contrast to previous studies, this study uses the *Deep Neural Network* (*Frozen*

Graph) method for face recognition. This research contributes to testing variations in distance and position so it can recognize a person's face even when wearing a mask and glasses. The input data for this research are facial images from *webcams* in *real-time.* To identify a face image, a face detection method is needed. This study uses two kinds of face detection, namely *Haarcasde* and *MTCNN*. This study also applies various variations of position and distance testing on real-time facial images. The reason for choosing the DNN algorithm is because this algorithm gives more precise and accurate results, as the DNN model is trained with large datasets and the model learns the best features from the dataset [20]. The reason for choosing *Haar Cascades* and MTCNN face detection is because they want to know which detection method is more suitable for use in the *Deep Neural Network* (*Frozen Graph*) method that has been created. According to research [21] the *Haar Cascades* can detect faces with *frame rates* highest and lightest.

2. METHODS

2.1 Load Dataset

The collection *datasets* form of images was taken from 25 Civil Servants in the Finance Bureau of the Ministry of Education and Technology, as many as 1853 face images, with the extension *.jpg*. Figure 1 shows the face images dataset.

Figure 1 Face images dataset

2.2 Augmentation and Preprocessing

This study's augmentation uses a 5° and 10° degrees rotation to the left and right. The translation is a shift with a significant distance of 3 and 6 pixels to the right, left, up, and down. And the addition of a brightness grayscale of 10, 30, and 45 is a lighter and darker technique. The preprocessing in this study is in the form of label encoding. The split dataset is used for training data as much as 85%, and 15% is used for testing and reshaping data.

2.3 Convolutional Neural Network (CNN) Modeling and Training

Generally, the CNN architecture has two stages, namely the feature learning and classification stages. Figure 2 shows feature learning stages, and Figure 3 shows classification stages. Feature learning is a technique that allows a system to run automatically to determine the representation of an image into features in the form of numbers representing the image. While the Classification Stage is a stage where the results of feature learning will be used for the classification process based on the predetermined subclass.

Figure 2 Feature Learning Stages

The first step in this research is to input the image shape using an image measuring 50x50x1. The number one in question is an image with one channel, namely a grayscale image. The next stage is the convolution process. Image is a matrix containing pixel values. Thus, in the first layer (convolution layer), a convolution process is carried out between the matrix of the input image and the kernel or filter with a particular matrix order and matrix value. Figure 2 illustrates the dimensions of the input image, namely (50 x 150 x 1). That means the face image has 50 pixels' row x 50 pixels' column x 1 channel (grayscale). The matrix is convoluted with four kernels or a filter with a size of 3x3 because the convolution process in this design has four layers of convolution. The result of the image convolution process is called a feature map. The calculation of the convolution operation is shown in equation (1).

$$\text{feature map} = (W-N+2P)/S+1 \tag{1}$$

The next stage is pooling or subsampling, which reduces the dimensions of each feature map resulting from the convolution operation. Several types of pooling layers include MaxPooling, AvgPooling, and SumPooling. In MaxPooling (e.g. kernel 2x2), the most significant value is taken for each element of the first 2x2 matrix in the convoluted feature map. The process runs to all aspects of the feature map (convolution results). At this stage, the activation process is carried out using the ReLU (Rectified Linear Unit) activation method because it can speed up the training process and is the most widely used. We can do the convolution, pooling, and activation methode repeatedly with different variations on the application of the CNN algorithm. The goal is to increase the extraction process so that the CNN algorithm that will be made can learn more and be more intelligent.
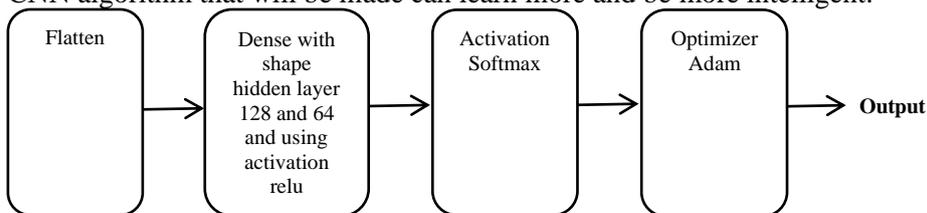

Figure 3 Classification Stages

Flattening converts a feature map into an one-dimensional array, which we will insert into the Neural Network (Dense) layer. Furthermore, the array will be the input for the neural network layer. The last layer is a Dense or Fully Connected Layer with numbers representing the number of labels or classes. In the case that only has one node output, we will use the sigmoid activation function. Whereas in this study, because the number of labels is more than one, the softmax activation function is used, which is very suitable for multi-class classification. The model created in this study uses the Adam optimizer (Adaptive moment estimation). We chose the Adam optimizer because it can find the position of the loss function that is very small, identifying minor errors (local minima) very quickly. For loss, use categorical cross entropy because the case of this research is a case of multi-class classification. And the matrix uses accuracy, precision, and recall. We must first set the epoch and batch size values to train the model.

2.4 Evaluate Model

We obtain a matrix plot at the *evaluation* results *training*. It is displayed as *a history accuracy plot* and *confusion matrix* to show how successful the *testing* was. If the *confusion matrix* results get a high value distributed on the diagonal from the top left to the bottom right, then the resulting model is said to be good.

2.5 Convert to OpenCV DNN

This stage is changing from .h5 extension to .pb extension using frozen graph technique. One of the advantages of the OpenCV DNN module is that it is highly optimized for

Intel processors. OpenCV DNN can achieve good and high FPS when performing inference on real-time video for object detection and image segmentation applications that have been trained using a specific framework.

2.6 Implementation of Transfer Learning

Transfer Learning is the reuse of previously trained models on new problems. In transfer learning, machines exploit knowledge gained from previous tasks to increase generalizations about others.

2.7 Identification of Test Data

At this stage, we test the data using the CNN model. Data is displayed on a page or frame. Then, the data containing an image of a person's face is labeled, and confidence results from face recognition.

2.8 Tests

Tests are conducted to determine the value or level of accuracy, precision, recall, and various image retrieval methods using the proposed model. In this study, the test was carried out by comparing some of the predicted data (data from the classification stage) with a set of actual data (data from the labeling stage) and taking facial image variations for recognition using the Deep Neural Network (Frozen Graph) method. We use Haar Cascade and MTCNN to identify which detection is more suitable for this method.

2.8.1 Testing of Accuracy, Precision, and Recall

Accuracy is the level of closeness between the predicted values. Precision is the level of accuracy between the information requested and the answers given by the equation system. The recall is the system's success rate in rediscovering a piece of equation information. The accuracy formula is shown in equation (2). The precision formula is shown in equation (3). The recall formula is shown in equation (4).

$$Accuracy = (TP+ TN)/(TP+TN+FP+FN) \tag{2}$$
$$Precision = TP/(TP+FP) \tag{3}$$
$$Recall = TP/(TP+FN) \tag{4}$$

2.8.2 Variation

Distance testing is conducted to determine whether the DNN model that has been created can recognize facial images displayed in *real-time*. This distance test is carried out by taking the distance between the face image and the *webcam* as far as 30 cm, 50 cm, 100 cm, and 200 cm. Testing of variations in the position of taking facial images with a webcam with variations in positions facing forward, looking up and down by 45°, looking left and right at 30°, 45°, and 90°, tilting right-left by 30°, and 45°, and taking facial images using a webcam with variations in the use of glasses and face masks. The accuracy variation formula is shown in equation (5).

$$accuracy_{variation} = \frac{sum\ true\ variation}{Total\ test} \tag{5}$$

3. RESULTS AND DISCUSSION

Implementation of the method uses seven main stages: data collection, augmentation and preprocessing, CNN modeling and training, evaluation model, Convert to OpenCV DNN, Implementation transfer learning, and identification of test data.

3.1 Load Dataset Stage

This objective is to find the position of the face of the image. Face images are detected

using the Haar Cascade method. Then, the face image that has been detected is resized to 50 × 50 pixels. After that, it converted to the grayscale format.

3.2 Augmentation and Preprocessing

Augmentation and preprocessing stages include rotation, translation, brightness grayscale, label encoding, dataset split, and data reshaping.

3.2.1 Augmentation Stage Rotation, Translation, Brightness Grayscale

Table 1 shows the augmentation process. The augmentation stage includes rotation, translation, and brightness grayscale. The result shows differences in the image before and after the operation.

Table 1 Augmentation Process

No	Label	Process	Before	After
1	Yogi Angga Putra	Rotation		
2	Yogi Angga Putra	Translation		
3	Yogi Angga Putra	Brightness Grayscale		

3.2.2 Modeling

The image processed at the convolution layer with a certain size_kernel will produce a feature map with smaller dimensions and the number of kernels. In Figure 2, the initial image has dimensions (*M x N x channel*) = 50 x 50 x 1 (grayscale). So, the calculation of block 1 from image convolution with N=3 is 64.

feature map $= (W-N+2P)/S+1$
$= (50-3+2(0))/1+1$
$= 48 \times 48 \times 64$

In block 2, convolution operations are performed as in block 1. This time the number of filters is 64 with a matrix of order (3 x 3). Here is the calculation:

feature map $= (W-N+2P)/S+1$
$= (48-3+2(0))/1+1$
$= 46 \times 46 \times 64$

```
Model: "sequential_1"
_____
Layer (type)                Output Shape            Param #
=========================================================
conv2d_1 (Conv2D)           (None, 48, 48, 64)      640

conv2d_2 (Conv2D)           (None, 46, 46, 64)      36928

max_pooling2d_1 (MaxPooling2 (None, 23, 23, 64)     0

conv2d_3 (Conv2D)           (None, 21, 21, 128)     73856

conv2d_4 (Conv2D)           (None, 19, 19, 128)     147584

max_pooling2d_2 (MaxPooling2 (None, 9, 9, 128)      0
_____
```

Figure 4 The results of the Layer Feature Map

Feature map enters the pooling layer with type MaxPooling, kernel matrix (2 x 2). The calculation is carried out as many as the commands initialized before, resulting in a value shown in Figure 4. So, the dimensions of the MaxPooling image are one-half of the feature map resulting from the convolution operation. The results of the pooling layer produce an image with dimensions of 23 x 23 x 64.

The last process in this modeling is training the model with the previously shared dataset. Figure 5 shows the results of the CNN experimental training model 1. In this training model, ten iterations (epochs) were carried out with time duration (ETA: 83 minutes or 1 hour 23 minutes) using the Jupiter lab.

The model in this experiment obtained an accuracy value of 0.9948 for data validation and a 0.0273 validation loss. Thus, we will be implement this CNN model in face image identification.

```
Train on 28114 samples, validate on 4962 samples
Epoch 1/10
28114/28114 [==============================] - 507s 18ms/step - loss:
0.4644 - accuracy: 0.8895 - val_loss: 0.0724 - val_accuracy: 0.9780
Epoch 2/10
28114/28114 [==============================] - 498s 18ms/step - loss:
0.0603 - accuracy: 0.9821 - val_loss: 0.0536 - val_accuracy: 0.9837
Epoch 3/10
28114/28114 [==============================] - 513s 18ms/step - loss:
0.0322 - accuracy: 0.9901 - val_loss: 0.0912 - val_accuracy: 0.9784
Epoch 4/10
28114/28114 [==============================] - 505s 18ms/step - loss:
0.0392 - accuracy: 0.9899 - val_loss: 0.0688 - val_accuracy: 0.9835
Epoch 5/10
28114/28114 [==============================] - 503s 18ms/step - loss:
0.0342 - accuracy: 0.9899 - val_loss: 0.0425 - val_accuracy: 0.9885
Epoch 6/10
28114/28114 [==============================] - 499s 18ms/step - loss:
0.0285 - accuracy: 0.9925 - val_loss: 0.0535 - val_accuracy: 0.9879
Epoch 7/10
28114/28114 [==============================] - 501s 18ms/step - loss:
0.0197 - accuracy: 0.9949 - val_loss: 0.0330 - val_accuracy: 0.9913
Epoch 8/10
28114/28114 [==============================] - 500s 18ms/step - loss:
0.0210 - accuracy: 0.9945 - val_loss: 0.0355 - val_accuracy: 0.9905
Epoch 9/10
28114/28114 [==============================] - 498s 18ms/step - loss:
0.0218 - accuracy: 0.9950 - val_loss: 0.0240 - val_accuracy: 0.9938
Epoch 10/10
28114/28114 [==============================] - 502s 18ms/step - loss:
0.0179 - accuracy: 0.9955 - val_loss: 0.0273 - val_accuracy: 0.9948
```
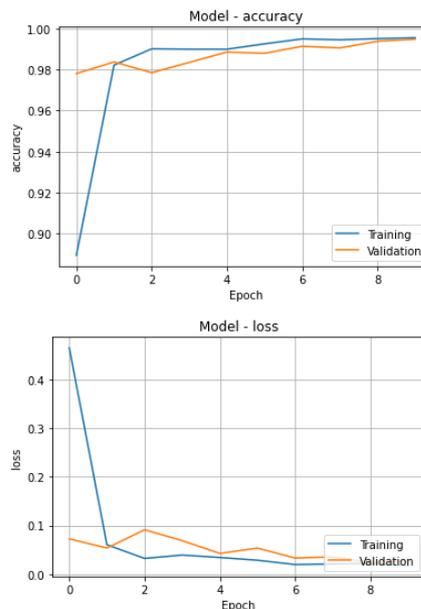
Figure 5 The Training Model

Figure 6 Plot History Accuracy

3.2.3 Evaluate Model

At this stage, the results of the CNN evaluation model we created will be displayed as a history plot of accuracy and confusion matrix. Figure 6 shows the details of the visualization of the CNN model evaluation results.

Representation in the confusion matrix of the overall prediction results of the test data contains 25 labels and 1853 face images. The actual label column (valid title) is the label data obtained through the labeling process. In contrast, the predicted label is the label data resulting from the classification process using the CNN model that has been created. Figure 7 shows the confusion matrix.
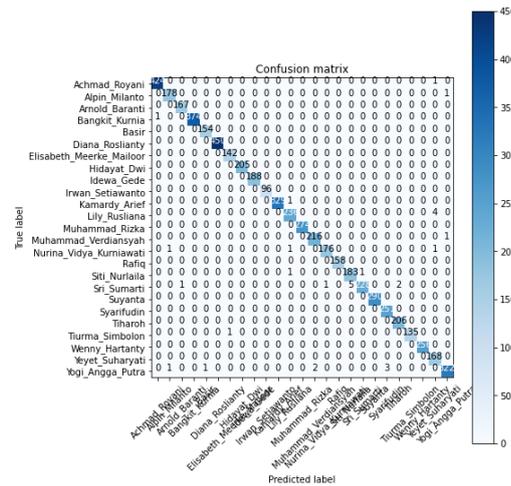
Figure 7 Confusion Matrix

3.2.4 Convert to OpenCV DNN

In the Convert to OpenCV DNN stage, the extension changes from .h5 to frozen graph model format. Figure 8 shows the details of the Convert to OpenCV DNN stage.
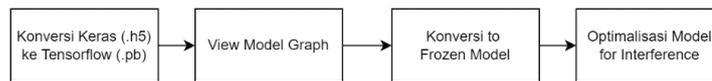

Figure 8 Stage Convert to OpenCV DNN

The first step is to convert the hard model format (.h5) that we created earlier into TensorFlow form (.pb). It is changed from variable to constant and sorted by displaying node name and node op. Then, save the result into the file model.pb.

3.2.5 Implementation Transfer Learning

The Implementation Transfer Learning stage occurs when we insert a new face image into the previously created model. Figure 9 shows an overview of the Implementation Transfer Learning stage.
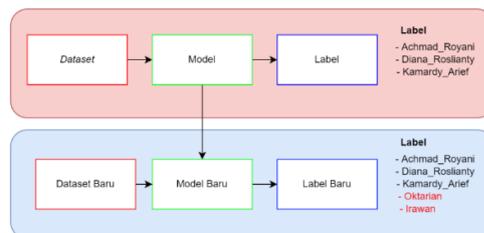

Figure 9 Implementation Transfer Learning Phase

3.2.6 Identification of Test Data

The first step in the identification phase of test data is loading the CNN model from the modeling results with the extension frozen inference graph (.pb). Next, initialize the label with the names of 25 civil servants' faces. The image will go through the process of changing the image size to 50 x 50, converting it to a grayscale image, setting a blob from the image for reading the DNN format, and changing shape. The image is converted to an array so that the image has dimensions (num image x num channel x height x width). Testing is successful when the model reads test data clearly with these dimensions. After the test image has been preprocessed, the image is ready to be identified by the CNN model. The model will display the label (face image name) and confidence.

Figure 10 Samples Figure of Test Data Identification Results

In this study, testing was carried out in terms of accuracy, precision, and recall on implementing the Convolutional Neural Network (CNN) algorithm in predicting or identifying labels for test data. Figure 10 shows samples figure of test data identification results. Its brands with Idewa_Gede, Diana_Roslianty, and Kamardy_Arief have 100% confidence that the model is ready for testing. Next will be testing variations in distance, position, and use of accessories.

3.3 Accuracy, Precision, and Recall Testing Using the CNN

Figure 11 shows model results of Accuracy, Precision, and Recall Testing. In implementing the *Convolutional Neural Network* (CNN) algorithm in predicting or identifying labels for test data, the model accuracy value is 99%, model precision is 99%, and 98% recalls.
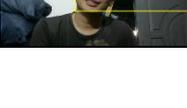
3.4 Testing for Variations in Distance, Position, and Use of Accessories

Tests for variations in distance, position, and use of facial accessories were conducted using *OpenCV Haar Cascades* and MTCNN detection. Table 2 shows the different results with OpenCV Haar Cascades and MTCNN detection. It contains the tests carried out for system development in evaluating, analyzing, and knowing the level of accuracy or similarity of results that have been achieved by the system that has been designed.

	precision	recall	f1-score	support
Achmad_Royani	1.00	1.00	1.00	425
Alpin_Milanto	0.99	0.99	0.99	179
Arnold_Baranti	0.99	1.00	1.00	167
Bangkit_Kurnia	1.00	1.00	1.00	375
Basir	0.99	1.00	1.00	154
Diana_Roslianty	1.00	1.00	1.00	450
Elisabeth_Meerke_Mailoor	0.99	1.00	1.00	142
Hidayat_Dwi	1.00	1.00	1.00	205
Idewa_Gede	1.00	1.00	1.00	188
Irwan_Setiawanto	1.00	1.00	1.00	96
Kamardy_Arief	1.00	1.00	1.00	330
Lily_Rusliana	0.99	0.98	0.99	240
Muhammad_Rizka	1.00	1.00	1.00	274
Muhammad_Verdiansyah	0.99	1.00	0.99	217
Nurina_Vidya_Kurniawati	0.99	0.98	0.99	179
Rafiq	1.00	1.00	1.00	158
Siti_Nurlaila	0.97	0.99	0.98	185
Sri_Sumarti	1.00	0.96	0.98	237
Suyanta	1.00	1.00	1.00	290
Syarifudin	0.99	1.00	0.99	251
Tiharoh	0.99	1.00	1.00	206
Tiurma_Simbolon	1.00	0.99	1.00	136
Wenny_Hartanty	1.00	1.00	1.00	256
Yeyet_Suharyati	0.97	1.00	0.98	168
Yogi_Angga_Putra	0.99	0.98	0.99	329
accuracy			0.99	5837
macro avg	0.99	0.99	0.99	5837
weighted avg	0.99	0.99	0.99	5837

Figure 11 Model Results of Accuracy, Precision, and Recall Testing

Table 2 Different Results with OpenCV Haar Cascades and MTCNN Detection

No.	Variations Testing	OpenCV *Haar Cascades*	MTCNN
1.	Distance ±30 cm		
2.	Distance ±50 cm		
3.	Distance ±100 cm		
4.	Distance ±200 cm		
5	Position look forward		
6	Position ±30º face up		
7	Position ±30º face down		
8	Position ±30º look right		
9	Position ±30º look left		
10	Position ±45º look right		
11	Position ±45º look left		
12	Position ±90º look left		
13	Position ±90º look right		
14	Position ±90º right tilt		
15	Position ±30º left tilt		

16	Position ±45º right tilt		
17	Position ±45º left tilt		
18	Accessories glasses		
19	Accessories Masks		

Testing for variations in the distance was carried out with a webcam as far as 30 cm, 50 cm, 100 cm, and 200 cm. We test data in variants of position face. Variants position face include look forward, face up and face down by 30°, face looking to the right and left by 30°, 45°, and 90°, and face tilting to the right and left by 30° and 45°. Testing for accessories is done by taking variations in the use of glasses and face masks. Table 3 shows the test results of various interpretations.

Table 3. Test Results of Various Interpretations

No	Type	*Haar Cascades*	MTCNN	Remarks
1	Testing for Distance Variations	100%	100%	The best distance is ±50 cm with *confidence* 99.9%
2	Testing for Position Variations	85%	96%	The best position is facing forward with *confidence* in both detections of more than 98%
3	Testing of Variations in the Use of Accessories	100%	100%	The best results on the use of masks with *confidence* 100%

The results of the distance variation test with the best results are with a distance of ± 50 cm using MTCNN detection. Meanwhile, the position variation test found that the MTCNN detection got 96% accuracy, and the best position was facing forward.

4. CONCLUSIONS

We successfully identified facial recognition in the present system with a webcam in real-time using the Deep Neural Network (DNN) algorithm. The results of the recognition test are to determine the ability of the system to recognize faces or not with various variations in distance, position, and use of masks and glasses. The percentage of distance variation testing using Haar Cascades detection is 100% and 100% MTCNN. The results of testing the position variation of Haar Cascades are 85% and MTCNN 92%. The percentage of testing variations in the use of masks and glasses for each face detection can recognize facial images up to 100%. The presence system using facial recognition is built using the Deep Neural Network (DNN) method, which it can do in real-time. With the DNN model, the data training process has been carried out ten times and got the best model with a validation accuracy value of 99.48% and a loss of 0.0273.

REFERENCES

[1] D. Jain, A. Upadhyay, A. Nirban, M. Arya, and R. Mishra, "Face Mask Detection & Attendance System," *Int. J. Sci. Res. Publ.*, vol. 11, no. 3, pp. 291–292, 2021, doi: 10.29322/ijsrp.11.03.2021.p11140.

[2] H. Fathi, F. Dai, and M. Lourakis, "Automated as-built 3D reconstruction of civil infrastructure using computer vision: Achievements, opportunities, and challenges," *Adv. Eng. Informatics*, vol. 29, no. 2, pp. 149–161, 2015, doi: 10.1016/j.aei.2015.01.012.

[3] W. Team, "WHO COVID-19 Dashboard," 2020. .

[4] W. TEAM, "WHO-Convened Global Study of Origins of SARS-CoV-2:China Part," 2021.

[5] F. inda N. Abida, I. W. Pastika, and Y. Rahman, "Analyzing the Text of the Regulation on Quarantine During the COVID-19 Pandemic : Forensic Linguistic Study," vol. 3, no. 2, pp. 221–228, 2022.

[6] M. Aritonang, I. D. Hutahaean, H. Sipayung, and I. H. Tambunan, "Implementation of Fingerprint Recognition Using Convolutional Neural Network and RFID Authentication Protocol on Attendance Machine," *ACM Int. Conf. Proceeding Ser.*, pp. 151–156, 2020, doi: 10.1145/3397391.3397449.

[7] S. Bhattacharya, G. S. Nainala, P. Das, and A. Routray, "Smart Attendance Monitoring System (SAMS): A Face Recognition based Attendance System for Classroom Environment," *2018 IEEE 18th Int. Conf. Adv. Learn. Technol.*, pp. 358–360, 2018, doi: 10.1109/ICALT.2018.00090.

[8] K. H. Teoh, R. C. Ismail, S. Z. M. Naziri, R. Hussin, M. N. M. Isa, and M. S. S. M. Basir, "Face Recognition and Identification using Deep Learning Approach," *J. Phys. Conf. Ser.*, vol. 1755, no. 1, 2021, doi: 10.1088/1742-6596/1755/1/012006.

[9] P. Gupta, N. Saxena, M. Sharma, and J. Tripathi, "Deep Neural Network for Human Face Recognition," *Int. J. Eng. Manuf.*, no. January, pp. 63–71, 2018, doi: 10.5815/ijem.2018.01.06.

[10] K. Alhanaee, M. Alhammadi, N. Almenhali, and M. Shatnawi, "Face Recognition Smart Attendance System using Deep Transfer Learning," *Procedia Comput. Sci.*, vol. 192, pp. 4093–4102, 2021, doi: 10.1016/j.procs.2021.09.184.

[11] G. Kaur *et al.*, "Neuroscience Informatics Face mask recognition system using CNN model," *Neurosci. Informatics*, vol. 2, no. 3, p. 100035, 2022, doi: 10.1016/j.neuri.2021.100035.

[12] A. Nurhopipah, I. R. Azziz, J. Suhaman, and U. A. Purwokerto, "Behind the Mask : Detection and Recognition Based-on Deep Learning," vol. 16, no. 1, pp. 67–78, 2022.

[13] J. S. Talahua, J. Buele, and P. Calvopiña, "Facial Recognition System for People with and without Face Mask in Times of the COVID-19 Pandemic," pp. 1–19, 2021.

[14] A. K. Sharadhi, V. Gururaj, S. P. Shankar, M. S. Supriya, and N. S. Chogule, "Face mask recogniser using image processing and computer vision approach," in *Global Transitions Proceedings*, 2022, vol. 3, pp. 67–73, doi: 10.1016/j.gltp.2022.04.016.

[15] R. E. Saragih and Q. H. To, "A Survey of Face Recognition Based on Convolutional Neural Network," *Indones. J. Inf. Syst.*, vol. 4, no. 2, pp. 122–139, 2022.

[16] M. Ranjan, V. Tripathi, and A. Dubey, "A novel data augmentation approach for mask detection using deep transfer learning," *Intell. Med.*, vol. 5, p. 100037, 2021, doi: 10.1016/j.ibmed.2021.100037.

[17] P. Gupta, V. Sharma, and S. Varma, "A novel algorithm for mask detection and recognizing actions of human," *Expert Syst. Appl.*, vol. 198, pp. 1–10, 2022, doi: 10.1016/j.eswa.2022.116823.

[18] A. Nurhopipah and A. Harjoko, "Motion Detection and Face Recognition For CCTV Surveillance System," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 12, no. 2, pp. 107–118, 2018, doi: 10.22146/ijccs.18198.

[19] A. Nurhopipah and U. Hasanah, "Dataset Splitting Techniques Comparison For Face Classification on CCTV Images," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 14, no. 4, pp. 341–352, 2020.

[20] N. Prasad, B. Rajpal, K. K. Rao Mangalore, R. Shastri, and N. Pradeep, "Frontal and Non-Frontal Face Detection using Deep Neural Networks (DNN)," *Int. J. Res. Ind. Eng.*, vol. 10, no. 1, pp. 9–21, 2021.

[21] M. Arora, S. Naithani, and A. S. Areeckal, "A web-based application for face detection in real-time images and videos," *J. Phys. Conf. Ser.*, vol. 2161, no. 1, 2022, doi: 10.1088/1742-6596/2161/1/012071.