Available online at http://scik.org

Commun. Math. Biol. Neurosci. 2023, 2023:19

https://doi.org/10.28919/cmbn/7872

ISSN: 2052-2541

FACE RECOGNITION FOR SMART ATTENDANCE SYSTEM USING DEEP LEARNING

GALUH PUTRA WARMAN, GEDE PUTRA KUSUMA*

Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia

Abstract: Attendance systems using traditional methods take much time and are less efficient. Face recognition is applied by comparing and recognizing the faces in the room with those in the database. However, we can address this issue by using a smart attendance system using face recognition to do attendance automatically and can save time in registering attendance. In this study, we propose a smart attendance system using the face detection model RetinaFace and MTCNN, then the face recognition model using FaceNet and ArcFace. This model will be evaluated using the WiderFace, Essex Faces 94, and Essex Faces95 datasets to evaluate its accuracy and speed. The final model results show that RetinaFace face detection has average precision (AP) results on the WiderFace dataset of 94.20% easy, 93.24% medium, and 83.55% hard, better than MTCNN with AP values of 83.31% easy, 80.32% medium, 83.55% hard. For the combination of face recognition models, FaceNet + RetinaFace obtained as the best combinations model in recognition and speed with a Rank-1 Recognition Rate of 99.114% and recognition speed per image of 118.90 ms.

Keywords: face detection; face recognition; deep learning; attendance system.

2020 AMS Subject Classification: 68T05, 68T45, 68T10.

*Corresponding author

E-mail address: inegara@binus.edu

Received January 03, 2023

1. INTRODUCTION

The attendance system was previously managed using traditional methods nowadays, so its implementation can be time-consuming and impractical. A smart attendance system with a combination of face detection and face recognition methods can overcome the ineffectiveness of attendance. Therefore, the introduction of image processing techniques to perform face recognition in an attendance system where the appearance of a student's face can be labeled and compared with images in the classroom with an accumulated database Naufal et al. [1]. Using face recognition smart attendance system automatically marks students attendance in a class by recognizing their faces. The system is divided into several steps, but face detection and face recognition are the main steps. First, we need a database on each face to mark attendance. The camera device takes all facial images in the classroom, then face detection is carried out, and then face recognition is carried out on the faces detected earlier by comparing the faces in the database containing images. By using a smart attendance system, it can reduce attendance errors assisted by using an automatic attendance list using a smart attendance system, making it easy for the parties concerned to use smart attendance automatically and can save time in registering attendance.

Huang et al., over the last ten years in the use of deep learning, have obtained interesting results from various fields, one of which is face recognition. The use of deep learning to perform face detection and face recognition in recent years has been widely applied in everyday life based on algorithms that perform learning on a lot of data by studying many factors such as faces, expressions, angles, and light Qu et al. [2]. There are several methods of face detection have been proposed, proposed by Deng et al. [3] RetinaFace method to perform face detection, which achieves stable face detection with evaluation results of average precision of 96.713%, 96.082%, and 91.44% on the WiderFace dataset. Ren et al. [4] used the multi-task cascade (MTCNN) method to perform face detection on pedestrians' feet. The evaluation results showed a detection rate of 74.1% for near-distance pedestrians, 66.8% for medium distance, and 34.2% for long distance. Proposed method YOLO V3 for face detection to perform with a yield of 92.79% Chen et al. [5].

The proposed method by Naufal et al. [1] uses the Haar feature detection algorithm, which

functions as face detection and feature extraction on images to obtain input with an accuracy rate of 95.23% in the synthesis dataset. There are methods for face recognition, William et al. [6] proposed the FaceNet method for performing face recognition, using MTCNN to take image face areas and then training them on the pre-trained model and evaluating face recognition on the Yale database, Jaffe and AT&T datasets with 100% results, 97.5%, and 100%. Proposed a face recognition method for students using Additive angular margin loss (ArcFace) on students, by using MTCNN for face extraction and entering into the ArcFace model trained with ResNet50 Jie et al. [7]. The results were evaluated on the LFW dataset with a face recognition accuracy rate of 99.80%. Proposed a method with the GoogleNet model (inception) using caffee and Nvidia digits framework to perform face recognition with an accuracy rate of 91.43% on the LFW dataset Anand et al. [8]. Proposed the SeNet50 model with an accuracy rate of 72% for facial recognition and 75.3% for image sizes of 8 x 8 pixels to 24 x 24 pixels Massoli, Amato, and Falchi [9].

Based on recent studies above, there are proposed solutions for face detection and face recognition. Therefore, this study presents the approach to using face detection models RetinaFace and MTCNN and face recognition with FaceNet and ArcFace models for smart attendance system. The objective that will be carried out in this paper is to find the best combination of face detection and face recognition methods that produce the best accuracy and speed of the face detection and face recognition models. The contributions of this paper are evaluating several face detection and face recognition models for smart attendance systems and finding the best combinations model for other organizations which the best to use in smart attendance system.

2. RELATED WORKS

In this section, related works are divided into 3 sections: Face Detection, Face Recognition, and summary related works. For face detection and Face Recognition, we analyze which model has the best performance and is suitable for the implementation described in related works.

2.1. Face Detection

Deng et al. [3] conducted a study using RetinaFace to perform face localization. By using

RetinaFace, which combines the prediction of face squares and localization of 2D facial landmarks and 3D vertex regression in the image plane, then performs face localization to obtain localization details on the face so that it can integrate with 3D regression, then uses a pyramid network to obtain image input with an output of 5 feature map, after that get the features on a map of a specific scale and calculate the multi-task loss. The results of the metric evaluation are based on the AP on RetinaFace using the WIDER FACE dataset with the final results of 96.713%, 96.082%, and 91.44% for the average AP result of 55.02%.

Zhang et al. [10] proposed a Multitask Cascaded Neural Networks (MTCNN). The proposed MTCNN's purpose is to construct an avalanched structure and utilize it as material for multi-task knowledge to anticipate the position of the face in a coarse-to-fine way. MTCNN also seeks to connect two tasks. In its application, MTCNN can identify real-time events with high accuracy. The MTCNN model is composed of three networks. The first is the Proposal Network (P-Net), which serves to obtain the face and give the face some boundary boundaries. The Refine Network (R-Net) removes several bounding boxes on the face by calibrating them and leaving just an accurate bounding box. The last network is the Output Network (O-Net). The O-Net works differently than the previous layers in that it takes the RNet result in the form of a boundary box and divides it into three layers: the first layer for face probabilities in the box, the second layer for boundary coordinates in the box, and the last layer for the coordinates of the five face landmarks. The results were evaluated on the FDDB dataset with a result true positive rate (TPR) of 0.95.

Proposed a method to perform face detection using a CNN called FaceDetectNet. Model architecture based on YOLO and GoogleNet as an implementation in face detection applications using an iterative proposal clustering (IPC) algorithm with facial output formed by CNN and a 2-level 'weak pyramid' which is used to detect small and large images. Using the in-depth features of the top hidden CNN layer to form face sizes of various sizes to obtain semantic information and global context for detecting small faces, the FaceDetectNet method was trained and tested on the WIDER FACE detection benchmark with an average precision (AP) of 0.69 at the hard level, 0.82 on medium and 0.8 on easy level Gorbatsevich and Vizilter[11].

Proposed a model named AInnoFace to perform face detection with high performance using one-stage RetinaNet as a baseline with a single and combined unified network, backbone and neck network is used as a multi-scale convolutional feature using RestNet-152 as six levels. Pyramid features are the backbone network structure Hang et al. [12]. Feature Pyramid Network (FPN) generates 6-level feature maps from p2 to p7 for detection. Focal loss is used for binary classification and IoU Loss for regression. The evaluation results for face detection with the AInnoFace model based on the AP metric on the WIDER Face dataset with three validation subsets get the results of 0.965 easy, 0.957 medium, and 0.912 hard.

Propose a lightweight CNN-based architectural method for real-time face detection; the proposed architecture consists of 2 main modules, namely CNN as a backbone to extract facial features and multi-level detection to make predictions from various scales Putro, Kurnianggoro, and Jo [13]. The proposed detector has one stage to be trained and tested using the WIDER FACE image input with challenges which has more difficult image results than other datasets. They take a loss-balancing approach and tweak the training. The results of the evaluation of the method used based on Average Precision (AP) are carried out on the AFW dataset WITH the results of 99.34, Pascal face results of 98.60, WIDER FACE conducted with 3 levels containing many small faces each level obtained results of 0.883 easy, 0.863 medium and 0.717 hard and, FDDB with evaluation based on ROC (Receiving operating characteristics) with results of 0.97.

2.2. Face Recognition

Proposed face recognition using the FaceNet method. In this study, FaceNet implementation used 2 pre-trained models, namely CASIA-WebFace and VGGFace2, in the FaceNet training process by MTCNN (Multi-task Cascaded Convolutional Neural Networks) when face detection has been carried out face image size with an area of 182 pixels X 182 pixels, then model training is carried out using 2 pre-trained models, namely CASIA-WebFace and VGGFace2 to improve the accuracy of face recognition, image datasets that have been edited. Evaluation results of accuracy 98.9% and 100% on Yale, 100% on JAFFE, 97.5% & 100% AT&T, 100% on Georgia Tech, 99.37% on Essex_faces94, 99.65% and 100% on Essex_faces95,76.86% and 77.67% on

Essex_faces96, 100% on Essex_grimace William et al. [6].

Proposes Additive Angular Margin Loss (ArcFace) using the ArcFace loss function further to improve the discriminative face recognition model and training process, using the arc-cosine function to calculate the angle between the current feature and the target weight. Add an additive angular margin to the target angle and log the target back again by the cosine function. And doing scale logits is the same as in softmax loss for classification tasks. The evaluation results were carried out on several datasets, and LFW got the best accuracy value of 99.82% Deng et al. [14].

Proposed Face Recognition using a multiple distance facial Convolutional Neural Network (CNN) architecture using the CNN method as feature extraction and facial images with optimal distances to train to show the best performance Moon, Seo, and Pan [15]. Doing preprocessing data using interpolation and histogram equalization to extract the image, then using CNN to extract all the features on the face, then using the Euclidian distance, which is used to ensure the arrangement of the faces in the database; CNN consists of 5 layers where the original images are placed with different distances of different sizes. Included, not included in the CNN structure. The first layer is the input layer, the 2nd layer is the convolution layer, the 3rd layer is the sub-sampling layer, the 4th layer is the convolution layer, and the 5th layer is the sub-sampling as the data used in training per person using the total 9 faces. The results of research on performing face recognition using the Euclidean distance method by comparing it with facial images in the database with the results of using the face recognition method average 88.9% within 1-9 meters.

Propose a research method using a deep neural network combined with a new alignment algorithm, PCA, and Bayesian network to perform multi-view face recognition Zhao et al. [16]. Using 3 CNN differences to obtain feature vectors and carrying out 2 classification methods during the L2 regularization training process is used to reduce overfitting, the activation functions used are ReLU, PreLu, and ELU for comparison. The PCA algorithm is used for dimensionality reduction in features, the joint Bayesian method is used for vector comfort assessment in face recognition. The evaluation results carried out in face image recognition obtained in the CAS-PEAL dataset obtained an accuracy rate of 98.52%.

2.3. Summary Related Works

Based on the reviews that have been carried out on several face detection and face recognition methods above, RetinaFace by Deng et al. [3] and MTCNN by Zhang et al. [10] have better performance in the face detection and alignment stage with quite high accuracy results where the results are better than other face detection methods. For face recognition, the best models were obtained by FaceNet by William et al. [6] and ArcFace Deng et al. [14]. FaceNet acquires high verification accuracy results in the Essex faces dataset in face recognition. ArcFace Model also consistently outperforms the state-of-the-art, which has obtained comprehensive experiments with other methods.

Conclusions that have been obtained based on the reading. In this study, a combination of 2 face detection and face recognition sides will be evaluated, and two models RetinaFace and MTCNN models, will be evaluated for the face detection side. On the face recognition side, FaceNet and ArcFace models will be evaluated.

3. THEORY AND METHODS

In this section theory and method are divided into 3 sections for deep learning and 2 subsections of face detection and face recognition models that will be used in this study.

3.1 Deep Learning

Deep learning is a subset of learning based on algorithms stimulated by brain function and how it works in a structured manner. Deep learning in computers can produce results by training on previously available examples. This deep learning capability makes it possible to achieve high results in related learning [17] Convolutional Neural Network (CNN) is one of the deep learning algorithms that work by taking an input image and then convoluting it with a filter or kernel to extract features with a feature extractor and then assigning weights and biases from the resulting data. Study various aspects or objects in the image to distinguish one from another [18].

The CNN architecture consists of several CNN layers consisting of an input layer, convolution layer, pooling layer, activation function, fully connected layer, and output layer [19]. Division of

work modules on CNN, namely feature extractor & classifier. The feature extractor module gets input data into an image, which will then undergo a convolution process for each remaining pixel in an image. This convolution process is carried out repeatedly to get results in each convolution process. The final process in the convolution layer is that the features (feature vector) will be used in the classification module to read the patterns in the feature vector. The read pattern will then be classified in the classification module & put in the CNN sample prediction output. The components contained in this CNN can not only be able to perform classification but can also be implemented to perform face detection and face recognition.

3.2 Face Detection Model

Face Detection is used to detect faces in each image, in this study the face detection that will be examined is RetinaFace and MTCNN.

3.2.1 RetinaFace

The retina face consists of 3 main components: a pyramid network (FPN) feature, context modeling capacity, and cascade multi-task loss as shown in Figure 1 below. The first part is the FPN with a feature pyramid level from P2 to P6, where P2 to P5 is calculated from the output in the residual ResNet stage adjusted from C2 to C5. Convolution 3x3 calculates P6 at stride=2 at C5. C1 to C5 are pre-trained classifications that have been trained on the ImageNet-11k dataset. While p6 is randomly initialized. The second part is the context modeling capacity to strengthen the deformation convolutional network (DCN). This module above features mapping apart from 3x3 convolution. The third part is Cascade Multi-task Loss, which is a loss function that is used to improve the performance of face localization. A cascade regression approach with multi-task loss is applied with a loss head of 1x1, the convolution is spread to different feature maps of the dimension. The first context head is used as a bounding box on the regular anchor, and the second context head is used to predict the bounding box more accurately than regressed, where the Loss function used is called mesh regression loss with a combination of vertex loss and edge loss.

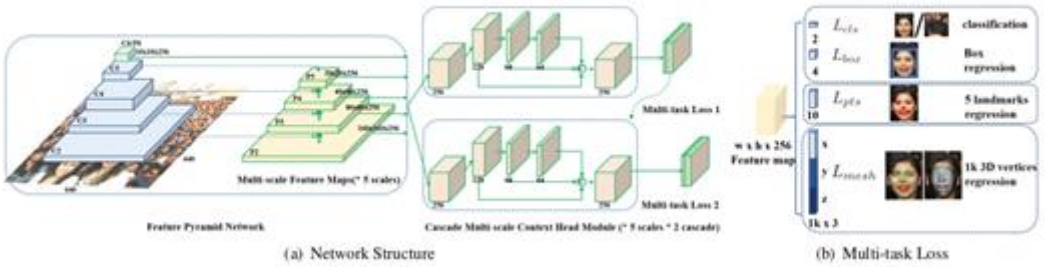

FIGURE 1. RetinaFace Architecture

3.2.2 MTCNN

Multi-Task Cascaded Convolutional Neural Networks (MTCNN) is a deep convolutional neural network-based face identification and alignment technique. MT (Multi-task), this approach may execute two tasks simultaneously: face detection and face alignment. MTCNN provides superior detection performance as compared to previous approaches. More correctly pinpoint the position of the face and fulfill real-time detection requirements.

MTCNN model, in the first step by giving an image, initially resize it to different scales to build an image pyramid, which is the input of the following three-stage cascaded framework name: PNet, RNet, and ONet. The original picture input is used to build the multi-scale image pyramid, which is then used to construct the candidate region via the complete convolution form of PNet. In RNet Non-maximum suppression is used to remove the candidate box with the highest degree of correspondence, then in ONet to identify face regions with more supervision. In particular, the network will output five facial landmarks positions as the structure diagram of the MTCNN model in Figure 2 below.

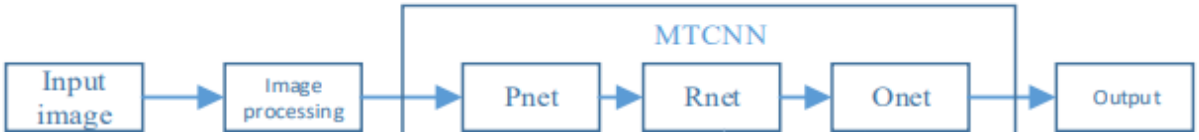

FIGURE 2. MTCNN Architecture

3.3 Face Recognition Model

Face Recognition is a method for identifying or verifying an individual's identity using their face. In this study, the face recognition models used are FaceNet and ArcFace.

3.3.1 FaceNet

FaceNet uses a deep convolutional network to optimize the embedding, which means comparing the bottleneck layers with the deep learning approach. FaceNet consists of part batch layers as input and deep architecture As CNN followed by l2 for normalization with embedding results, using triplet loss when applied in the training process as described in Figure 3. By using the one-shot learning method, FaceNet directly trains faces with Euclidean space which contains similarities between faces. FaceNet uses triplet loss in the training process by performing face-to-face matching to minimize the distance between positive anchors and maximize the negative anchor distance where these positive values have the same identity and negative have different values.

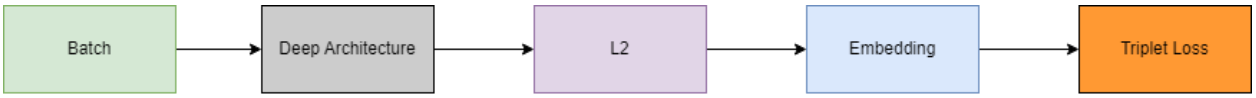

FIGURE 3. FaceNet Architecture

3.3.2 ArcFace

ArcFace is a face recognition model that takes two face images as input and outputs the distance between them to see how likely they are to be the same person. It can be used for face recognition. ArcFace uses a similarity learning mechanism that allows distance metric learning to be solved in the classification task by introducing Angular Margin Loss to replace Softmax Loss. The distance between faces is calculated using cosine distance, a method used by search engines, and can be calculated by the inner product of two normalized vectors. If the two vectors are the same, θ will be 0 and cosθ=1. If they are orthogonal, θ will be π/2 and cosθ=0. Therefore, it can be used as a similarity measure.

$$L3 = -\frac{1}{N}\sum_{i=1}^{N} log \frac{e^{s\left(cos\left(\theta_{y_i}+m\right)\right)}}{e^{s\left(cos\left(\theta_{y_i}+m\right)\right)} + \sum_{j=1, j\neq y_i}^{n} e^{scos\theta_j}} \qquad (1)$$

The Arcface loss function in Eq. (1) essentially takes the dot product of the weight ‘w’ and the ‘x’ feature where θ is the angle between ‘w’ and ‘x’ and then adds a penalty ‘m’ to it. ‘w’ is normalized using l2 norm and ‘x’ has been normalized with l2 norm and scaled by a factor ‘s’ This makes the predictions rely only on the angle θ or the cosine distance between the weights and the feature.

In a typical classification task, after calculating features, the Fully Connected (FC) layer takes the inner product of features and weights and applies Softmax to the output. In ArcFace, cosθ is calculated by normalizing features and FC layer weights and taking the inner product. The loss is calculated by applying Softmax to cosθ. At this point, apply arccos to the cosθ values after taking the inner product and add an angular margin of +m only for the correct labels. In this way, we prevent the weight of the FC layer from being overly dependent on the input data set. During the ArcFace inference process, the features of the two faces are normalized, and the inner product is computed to determine if both pictures are the same person.

4. PROPOSED METHODOLOGY

This study focuses on creating a smart attendance system that applies a combination of face detection models and face recognition combinations of face detection models, namely MTCNN by Zhang [10] and RetinaFace by Deng et al. [3]. Face recognition model to use in this study William et al. [6] and ArcFace Deng et al. [14]. The proposed methodology is illustrated in Figure 4 below.

FIGURE 4. Overview Proposed Methodology

Image input will go to the MTCNN or RetinaFace face detection models for finding and extracting faces from photos. After that, enter the face embedding using FaceNet or ArcFace for the vector that represents the features extracted for each detected face, and the classifier model using Multi-Layer Perceptron (MLP) that used will take a face embedding as input to predict

identity of a given face. The input image has been resized with two dimensions, 160 x 160 which is the size to fit into the FaceNet model, and the image size 112 x 112 to fit into the ArcFace model. Table 1 consists of 4 combinations models, the baseline used in this research is the combination model are MTCNN+FaceNet and MTCNN+ArcFace.

TABLE 1. Face Detection + Face Recognition Combination

No	Combination Models	
	Face Detection	Face Recognition
1	RetinaFace	FaceNet
2	RetinaFace	ArcFace
3	MTCNN	FaceNet
4	MTCNN	ArcFace

4.1 Dataset

The dataset used in this study consist of three dataset WiderFace for face detection models and combinations of Essex Faces94+95 to determine the best performing combinations method at face recognition models.

4.1.1 WiderFace

The WiderFace dataset will also be used for training and evaluation in the model used for the face detection process.WiderFace consist of 32,203 images with 393.703 annotated faces, 158.989 of which are in the train set, 3.227 in the validation set, and the rest in the test set. The validation are divided into easy, medium, and hard subsets respectively with sample WiderFace dataset in Figure 5 below.

Figure 5. Sample WiderFace

4.1.2 Essex Faces94

In this study, the Essex Faces94 dataset was used to evaluate the face recognition model with a total of 153 identities with an image size of 180 x 200 with a total of 3,080 images with subjects

sitting at approximately the same distance from the camera and was asked to speak while a sequence of twenty images was taken. The speech was used to introduce moderate and natural facial expression variations with an example sample dataset in Figure 6 below.

Figure 6. Sample Essex Faces94

4.1.3 Essex Faces95

The Essex Faces95 dataset in this study is used to evaluate the performance of the face recognition model with a total of 72 individual identities with an image size of 180 x 200 consist a total of 1.440 images. A sequence of 20 images per individual was taken using a fixed camera. During the sequence the subjects take one step forward towards the camera. This movement is used to introduce significant head (scale) variations between images of the same individual, as shown sample dataset in Figure 7 below.

Figure 7. Sample Essex Faces95

5. EXPERIMENTAL DESIGN

This section will explain the experiments that will be carried out on face detection evaluation and evaluation of the combination of the two models for measuring performance from recognition rate and speed.

5.1 Evaluation of Face Detection

In the Face Detection experiment, the dataset for WiderFace was divided into the following sizes: 40% for training, 10% for validation, and 50% for testing. With size training data running on MTCNN and RetinaFace face detection models. Furthermore, validation data is used for testing models because WiderFace does not provide ground truth evaluation for data testing. The first step of Experiments is to take image annotations from the training, then do the face detection model training, after that testing on the validation set by getting average value precision (AP) of 3 subset validation easy, medium, and hard in the result respectively.

5.2 Evaluation of Combination Models

In the combination models experiment, the dataset from EssexFaces94+EssexFaces95 is combined with a total of 4519 images with a total of 224 identities, split into 60% training, 20% validation, and 20% testing. The first experiment was performed by extracting facial images with the face detection models and looping to get images and identities from the dataset subfolders. Next, embedding is done to get facial features that will be input into the FaceNet or ArcFace face recognition models and stored in an array. After that, the embedding results are vector normalized with the l2 norm. Last step is to do a fit model by getting train, val, and test results using a multi-layer perceptron (MLP) classifier to get a rank-1 recognition rate.

5.3 Performance Measures.

This section will explain the evaluation of the metrics used to measure the performance of the face detection, face recognition, and speed models of the combination of the two models.

5.3.1 Face Detection Performance

The performance measurement used to evaluate the model's performance on face detection models is called average precision (AP), a method of condensing the precision-recall curve into a single value representing the mean of all precisions. The following Eq (2). below is used to calculate the AP. The difference between the existing and the next recalls is measured and then multiplied by the current precision and use a loop that goes through all precisions/recalls. In other words, the AP is the weighted sum of precisions at each n threshold, with the weight corresponding to the increase in recall.

$$AP = \sum_{k}^{k=n-1}[Recalls(k) - Recall(k+1)] * Precision(k) \quad (2)$$

5.3.2 Face Recognition Performance

The performance measurement used to evaluate the model's performance on face recognition is the rank-1 recognition rate. Which depends on a list of images from one gallery and a list of images to test with the same identity; for each probe image, all similarities to all images in the gallery folder will be calculated and determined if the gallery image has the highest or lowest similarity from the identity which is the same as the probe image, for the identification of the query image compared to a set of target images in the gallery then sorted by similarity to the most similar or the smallest most similar target image. With the formula Eq (3). below.

$$Rank - 1\ Recognition\ Rate = ((Total\ Correct\ identified\ images)/(Total\ of\ Images)) * 100\% \quad (3)$$

5.3.3 Processing Time Performance

The performance of the processing time is measured by an image that can pass processing time through detection, embedding, and recognition, which is performed on a combination of face detection and face recognition models to determine the best combined model based on processing time elapsed as shown formula in Eq (4) below.

$$Processing\ Time = Detect\ Time\ Elapsed + Embedding\ Time\ Elapsed + Recognition\ Time\ Elapsed \quad (4)$$

6. EXPERIMENTAL RESULT

In this study, the experimental results are divided into 3 parts evaluation results of face detection, evaluation results of combination models, and evaluation results of processing speed.

6.1 Evaluation Results of Face Detection

The experimental testing results on the WiderFace dataset on face detection models show that RetinaFace outperforms MTCNN with an AP value in Table 2.

TABLE 2. Evaluation Result of Face Detection Models

Model	Average Precision (AP) (%)		
	Easy	Medium	Hard
RetinaFace	**94.20**	**93.24**	**83.55**
MTCNN	83.31	80.32	58.72

6.2 Evaluation Results of Combination Models

The experimental results were carried out after combining RetinaFace and MTCNN for face detection models, with FaceNet and ArcFace predicted using the Multi-Layer Perceptron (MLP) classifier.

TABLE 3. Training and Validation Evaluation Result of Combination Models

Combination Models	Rank-1 Recognition Rate(%)	
	Training	Validation
RetinaFace+FaceNet	**99.341**	**99.003**
RetinaFace+ArcFace	98.416	94.795
MTCNN+FaceNet	99.189	98.780
MTCNN+ArcFace	98.747	93.902

As shown in Table 3. RetinaFace+FaceNet combination models produce better values than the other three models, with a training accuracy of 99.341%. Furthermore, the hyperparameter tuning is done can give optimized values for the models. With 3 different learning rates (LR) for the experiments of 0.1, 0.01, and 0.001. after conducting the experiment, the results obtained were the combination of the four best models with an LR value of 0.001 will be used to perform predictive accuracy models because it has best optimization values in validation result with 99.003% in RetinaFace+FaceNet..

TABLE 4. Testing Evaluation Results on Combination Models

Combination Models	Rank-1 Recognition Rate (%)
RetinaFace+FaceNet	**99.114**
RetinaFace+ArcFace	94.053
MTCNN+FaceNet	98.899
MTCNN+ArcFace	92.401

The results in Table 4. show that the testing results obtained for all combination models are quite high, RetinaFace+FaceNet holds the best recognition rate with a value of 99.114%.

6.3 Evaluation Results of Processing Speed

In this section, an experiment was conducted to measure the performance of a combination model based on speed with a total of 100 images measured 10 times, and the average was found

to obtain the processing speed per image.

TABLE 5. Speed Evaluation Results on Combination Models

Speed (per image)	
Combination Models	Average Processing Time (ms)
RetinaFace+FaceNet	118.90
RetinaFace+ArcFace	**104.98**
MTCNN+FaceNet	1016.58
MTCNN+ArcFace	1021.41

The results in Table 5. show that the RetinaFace+ArcFace model has the fastest speed in image processing for facial recognition with a value of 104.98 ms per image, for the RetinaFace+FaceNet model gets a slightly longer value.

Based on the results of the experiments that have been carried out, the RetinaFace+FaceNet combination model has a higher Rank-1 Recognition Rate than the other three models, in terms of speed RetinaFace+ArcFace has a faster time processing images in face recognition where slightly 13.92 ms faster than RetinaFace+FaceNet. Therefore, the recommendation for the best combination model based on the speed and higher accuracy that can be used is RetinaFace+FaceNet.

7. CONCLUSIONS

In this study, based on experiments conducted on face detection models with RetinaFace and MTCNN as measured by AP, this model will be combined with every FaceNet model and ArcFace facial recognition model as measured by Rank-1 Recognition Level. The results of the face detection model show that RetinaFace is the best model based on AP values of 94.20% easy, 93.24 % medium, and 83.55% hard.

The results of the face detection model combined with the face recognition model with a Rank-1 Recognition Rate measurement show that RetinaFace+FaceNet has the best performance with a value of 99.114% compared to the other three combination models. For processing time performance, the RetinaFace+ArcFace model has the fastest speed from the first step of face detection, face embedding, and face recognition with a processing time value of 104.98 ms, slightly

faster than RetinaFace+FaceNet with different time of 13.92 ms.

It can be concluded that by using the recommended combination of face detection and face recognition models, RetinaFace+FaceNet has advantages in terms of Recognition Rate and speed to be implemented in a smart attendance system.

In this study the evaluation results were run on sufficient hardware powerful, and the specification on this hardware are better than the system on chip Arduino and Raspberry Pi. Future Works it is necessary to evaluate how performance of this combination model with hardware at the same computing capacity lower like Arduino or Raspberry Pi. From this research, it has only been evaluated in terms of processing speed, Furthermore, it also needs to be evaluated from the needs of its computing resources.

CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests.

REFERENCES

[1] G.R. Naufal, R. Kumala, R. Martin, et al. Deep learning-based face recognition system for attendance system, ICIC Express Lett. Part B: Appl. 12 (2021), 193-199. https://doi.org/10.24507/icicelb.12.02.193.

[2] X. Qu, T. Wei, C. Peng, et al. A fast face recognition system based on deep learning, in: 2018 11th International Symposium on Computational Intelligence and Design (ISCID), IEEE, Hangzhou, China, 2018: pp. 289–292. https://doi.org/10.1109/ISCID.2018.00072.

[3] J. Deng, J. Guo, E. Ververas, et al. Single-shot multi-level face localisation in the wild, In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 5203-5212.

[4] L. Ren, W. Xian, H. Tang, et al. Pedestrian and face detection with low resolution based on improved MTCNN, in: Proceedings of the 2020 9th International Conference on Computing and Pattern Recognition, ACM, Xiamen China, 2020: pp. 174–180. https://doi.org/10.1145/3436369.3436492.

[5] W. Chen, H. Huang, S. Peng, C. Zhou, C. Zhang, YOLO-face: a real-time face detector, Visual Computer. 37 (2020), 805–813. https://doi.org/10.1007/s00371-020-01831-7.

[6] I. William, D.R. Ignatius Moses Setiadi, E.H. Rachmawanto, et al. Face recognition using FaceNet (survey, performance test, and comparison), in: 2019 Fourth International Conference on Informatics and Computing (ICIC), IEEE, Semarang, Indonesia, 2019: pp. 1–6. https://doi.org/10.1109/ICIC47613.2019.8985786.

[7] Y. Jie, M. Jiong, S. Baiyi, et al. Face recognition of students based on ArcFace, in: Proceedings of the 2020 International Conference on Aviation Safety and Information Technology, ACM, Weihai City China, 2020: pp. 678–681. https://doi.org/10.1145/3434581.3434712.

[8] R. Anand, T. Shanthi, M.S. Nithish, et al. Face recognition and classification using GoogleNET architecture, in: K.N. Das, J.C. Bansal, K. Deep, et al. (Eds.), Soft Computing for Problem Solving, Springer Singapore, Singapore, 2020: pp. 261–269. https://doi.org/10.1007/978-981-15-0035-0_20.

[9] F.V. Massoli, G. Amato, F. Falchi, Cross-resolution learning for face recognition, Image Vision Comput. 99 (2020), 103927. https://doi.org/10.1016/j.imavis.2020.103927.

[10] K. Zhang, Z. Zhang, Z. Li, et al. Joint face detection and alignment using multitask cascaded convolutional networks, IEEE Signal Process. Lett. 23 (2016), 1499–1503. https://doi.org/10.1109/LSP.2016.2603342.

[11] V.S. Gorbatsevich, A.S. Moiseenko, Y.V. Vizilter, FaceDetectNet: face detection via fully-convolutional network, Computer Optics. 43 (2019), 63-71. https://doi.org/10.18287/2412-6179-2019-43-1-63-71.

[12] F. Zhang, X. Fan, G. Ai, et al. Accurate face detection for high performance, (2019). https://doi.org/10.48550/ARXIV.1905.01585.

[13] M.D. Putro, L. Kurnianggoro, K.H. Jo, High performance and efficient real-time face detector on central processing unit based on convolutional neural network, IEEE Trans. Ind. Inform. 17 (2021), 4449–4457. https://doi.org/10.1109/TII.2020.3022501.

[14] J. Deng, J. Guo, N. Xue, Arcface: Additive angular margin loss for deep face recognition, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, p. 4690–9.

[15] H.M. Moon, C.H. Seo, S.B. Pan, A face recognition system based on convolution neural network using multiple distance face, Soft. Comput. 21 (2016), 4995–5002. https://doi.org/10.1007/s00500-016-2095-0.

[16] F. Zhao, J. Li, L. Zhang, et al. Multi-view face recognition using deep neural networks, Future Gen. Computer Syst. 111 (2020), 375–380. https://doi.org/10.1016/j.future.2020.05.002.

[17] A. Bashar, Survey on evolving deep learning neural network architectures, J. Artif. Intell. Capsule Networks

2019 (2019), 73–82. https://doi.org/10.36548/jaicn.2019.2.003.

[18] R. Chauhan, K.K. Ghanshala, R.C. Joshi, Convolutional neural network (CNN) for image detection and recognition, in: 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), IEEE, Jalandhar, India, 2018: pp. 278–282. https://doi.org/10.1109/ICSCCC.2018.8703316.

[19] L. Alzubaidi, J. Zhang, A.J. Humaidi, et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions, J. Big Data. 8 (2021), 53. https://doi.org/10.1186/s40537-021-00444-8.