Custom Deep Learning Face Recognition based on Tensorflow Model for Attendance Management System

Yosua Alvin Adi Soetrisno
Department of Electrical Engineering
Diponegoro University
Semarang, Indonesia
yosua@live.undip.ac.id

Aghus Sofwan
Department of Electrical Engineering
Diponegoro University
Semarang, Indonesia
asofwan@elektro.undip.ac.id

M. Arfan
Department of Electrical Engineering
Diponegoro University
Semarang, Indonesia
arfan@elektro.undip.ac.id

Sumardi
Department of Electrical Engineering
Diponegoro University
Semarang, Indonesia
sumardi@elektro.undip.ac.id

2021 5th International Conference on Informatics and Computational Sciences (ICICoS) | 978-1-6654-3807-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICICOS53627.2021.9651766

***Abstract*—Face recognition technology has many implementation roles in the attendance management system. Attendance systems need proper solutions to detect a face in real-time situations using a particular purpose device. Face recognition systems can differentiate human faces based on face features trained in the deep learning model. Although significant advances in face recognition can increase the variety of face conditions in detection, some challenges still exist in face recognition that makes the existing model need to be taken apart and reparametrized. Challenge comes from lighting condition, blurred condition, also face tilt position. This research design a superficial layer of convolutional neural network using a built-in Tensorflow sequential model library. Generally, a transfer learning mechanism is used in object detection, especially in face recognition. This research doesn't use transfer learning because the accuracy of an existing model like the InceptionV1 model gives good accuracy in cross-validation training but gives a significant error in testing the trained face. The attendance management system was built in Flask Web Framework because developed in a Python language environment. The accuracy of the custom model has an average of 88.23%, which is tested with 16 different students, with each student having 48 pictures.**

Keywords*—*face recognition, deep learning, convolutional neural network, TensorFlow, attendance management

I. INTRODUCTION

Face recognition technology has existed for a long time, but it was not very accurate to detect the face profile until today. Face recognition is used in many terms, such as secure authentication, attendance verification [1], and social media tagging in the posted picture. The advancement in facial recognition will use raw filtered layered features trained via a deep learning model. The human face has many differential features that could construct the semantic information of the people, like facial expression, facial structure, and facial topology. Traditional face recognition, such as the Eigenface, PCA [2][3], LDA [2], GABOR [2], Local Binary Pattern (LBP) [4], improved LBP [4][5], had some shortages because of limited feature extraction mechanisms. There is a lot of advancement and modification of this algorithm [4] to increase face detection accuracy. With the proper improvement of the traditional algorithm to get the more fit feature combined with deep learning, the face recognition model's accuracy could be improved. Deep Learning technology has dominated object detection in terms of performance and availability [6]. The transfer learning of Convolutional Neural Network (CNN) architecture has been implemented in many computer vision problems such as object detection.

In object detection, especially in face recognition, there is a strengthening of characteristic properties of the input given as a filter, weight, and activation function that could cut off the irrelevant properties. The extracted properties in the first network layer most probably indicate the presence or absence of the edges in specific directions and positions in the image. The deeper layer identifies patterns by detecting special edge arrangements by considering insignificance changes in edge locations [6]. The next hidden layer learns more complex shapes based on the edge previously known. This sequential structure allows deep learning to automatically find the convenient feature based on the filter and activation function systematically. This process simplifies the feature extraction process because there is no need to understand the most relevant part of the facial characteristic.

This research proposes a custom CNN architecture for the face recognition model used in attendance management. Custom CNN was built because the available deep learning model, such as MobileNetv2 [7] and FaceNet [8], could not detect the face has been trained very well. There are 18 hidden layers in the custom CNN architecture. The two characteristics of hidden layers that were used are stacking and producing [9]. Stacking is used to extract the feature with the convolutional, max pooling, and flattening type layer [10]. A fully connected layer type is used for feature matching production. A fully connected layer has its density based on the number of face classes in the "softmax" activation function and 1024 units representing several neurons used in the "relu" activation based on the input pixel convoluted in the stacking process. The dataset comes from student photos taken from any angle.

II. LITERATURE REVIEW

Various deep learning techniques done in the term of face recognition and verification such as Deep Models [9], DeepFace [6], DeepID3 [11], FaceNet [8], L2-Softmax [6], AttendXNet3 [11], SIAMESE [2], MobileNetV2 [12], ArcFace [13], MTCNN [14]. Deep Models research [9] explained joining many face recognition algorithms and mechanisms before deep learning. Feature extraction that was used was geometric and appearance-based [15]. Landmark estimation is state of the art in traditional face recognition using the DLIB library [16]. This mechanism is combined with selecting the region of interest followed by geometric features like LBP and Normalized Central Moment (NCM). Finally, traditional machine learning with a Support Vector Machine (SVM) algorithm for the classification task.

Deep learning is the new emerging algorithm used in face recognition. There is DeepFace [6], which is claimed as a near-human face recognition skill model, used AlexNet architecture with a "softmax" activation function trained with Facebook dataset and obtained an accuracy of 97.35% [11]. DeepID3 used VGGNet-10 [17] architecture used contrastive loss. Contrastive loss used contradiction in a distance between positive and negative examples outputs [11]. CelebFaces+ is a variation of the dataset used besides Facebook.

FaceNet trains CNN using Stochastic Gradient Descent with standard backpropagation and AdaGrad as the loss function optimization. Two convolutional layers are compared and tested [5]: traditional CNN with 22 hidden layers and Google-LeNet using the Inception model. The result shows that the validation rate with classic CNN could get an accuracy of 87.9% compared with Inception Google-LeNet which is 89.4%. The difference between the validation rate accuracy at that two models is not too far. This idea makes this research consider creating a custom simpler CNN rather than using transfer learning from a complex model.

L2-softmax used ResNet-101 architecture with level 2 regularization gained an accuracy of 99.78% [6]. L2-softmax's research shows that many models could be used as transfer learning for face recognition, but when tested, the accuracy goes random rather than offering a stable classification result. AttendXNetV3 used Resnet-34 architecture with fewer hidden layers to extract face features, but the classification used efficient similarity search. Faiss similarity and manhattan distance are used to measure the face feature and make a query based on an index created in the database pool to match the closest face feature. Database pool showed a variation of classification function from using a database to make distance calculation or another feature that makes the CNN more complex with another level of the network like in the SIAMESE.

SIAMESE [2] is a network of probability that could solve multiple sample input and classification. SIAMESE uses hierarchical training with two CNN. The first level of the network is used to face positioning, and the second level is used for facial landmark detection. MobileNetV2 [12] was used for fruit detection. MobileNetV2 extracted the fruit features and the activation function using "softmax" as a classifier. There is error optimization using Adam optimizer. The accuracy of fruit detection was 85.12%. This research considers choosing a basic model for classification using "softmax" because SIAMESE and Faiss need a more complicated feature like face landmark and database pool that is not very suitable with the available dataset and the attendance system itself. The attendance system in the other hand need a fast detection and this research then will be implemented in AI developer kit board like Atlas Hisense 200 DK, so needed more simple and lightweight model.

CCTV implementation at university was done in ArcFace [13] research. ArcFace model proposed a new loss function, an additive angular margin that could highly discriminate features of the face. ArcFace showed the best result among other loss functions such as triplet loss, intra-loss, and inter-loss. Arcface showed that modification of loss function could increase the accuracy.

MTCNN [14] used a deep cascaded multi-task framework using different features. This correlated model objects that exist in the face then measured the correlation strength between the elements. MTCNN [14] showed that it needed a better judgment in similarity using cascaded CNN [18].

TensorFlow's flexibility in designing highly modular models is a strong advantage. Many pieces of the Tensorflow building block must be known before building the model. That situation motivated this research to use the Tensorflow and Keras model. Developing high-level APIs like Keras and Slim, which look like an abstract in many parts required in creating machine learning algorithms, so the characteristic of each layer and neuron could intuitively match with the formula. Many researchers and developers use Tensorflow in this section. Many concerns and problems could be readily resolved because they are typically the same issues many other individuals face.

The contribution of this research is to design a custom model of CNN using Tensorflow and Keras model that could fit the entire limited dataset with no landmark. A complex model like MobileNetV2 and Google-LeNet transfer learning has been tried and showed not very good enough to detect a face. However, the cross-validation average accuracy is between 98-99% above. The convolutional layer is used with kernel and bias regularization parameters. Kernel regularizer tries to reduce the weights W, and bias regularizer tries to reduce the bias b. This parametrization combined with the "relu" activation function could implicitly solve a problem with solid statistical benefits.

III. Research Methodology

Deep learning is one branch of machine learning algorithm using the architecture of neural network with some advancement in a hidden layer, backpropagation, and memory arrangement. Deep learning could learn massive training data based on the image with a convolution process to get the feature using windowing. The deep learning convolutional layer has many filters that select the component from the most detail until the most available using window variation. This filter could capture the part that is not considered to be carried in traditional feature extraction.

A. Convolutional Neural Network

The evolution of CNN solved some weaknesses of neural networks like computational complexity and translational invariance. Translational invariance means that the network interprets input patterns the same way in CNN. The feature was captured as the feature filter that could be shifted in the windowing process. The challenge in CNN is that the neural network learns to memorize rather than understand when many samples have a similar characteristic, and the variation of the training object is not very wide. When this situation occurs, the CNN only understands the piece of the picture rather than the whole concept. CNN has a local receptive field, sharing weights, and spatial domain that could manage the invariance of scaling and distortion of the image [2].

The CNN has a convolutional layer and pooling layer that could capture the essential feature of the image in a lighter dimension. The feature captured then matched with the weight of each neuron in the fully connected layer. The convolutional layer has a shared weights formula in equation (1) [2].

$$C_{i,j,k}^{t} = g(\Sigma_{z=1}^{C_s}\Sigma_{y=1}^{w_c}\Sigma_{x=1}^{h_c}\mathrm{I}_{i+x-1,j+y-1,c_k^t(z)}^{t-1} * F_{x,y,k}^{t} + B_k) \quad (1)$$

Compared with unweights formula, there is a difference in the weighting mechanism shown in equation (2) [2].

$$C_{i,j,k}^{t} = g(\Sigma_{z=1}^{C_s,k}\Sigma_{y=1}^{w_c}\Sigma_{x=1}^{h_c}\mathrm{I}_{i+x-1,j+y-1,c_k^t(z)}^{t-1} * F_{i,j,x,y,k}^{t} + B_k) \quad (2)$$

c_s is the connection of the k-th unit in convolutional layer with the last layer. w_c is the width of a convolutional layer unit. h_c is the height of the convolutional layer unit. t is the number of the convolutional layer. C_k^t is the number of the k-th unit in convolutional layer with the last layer.

There is a repetition possibility between a combination of pooling layer after convolutional layer. The pooling layer worked separately among each feature map to make a new set of same-size feature maps. The pooling layer selects the critical feature that works like a filter. The size of the pooling operation or filter is smaller than the size of the feature map. For example, a pooling layer 2x2 (4 pixels) applied to a feature map of 6×6 (36 pixels) will result in an output pooled feature map of 3×3 (9 pixels). The pooling operation is stated in each hidden layer. There are two pooling mechanisms: average pooling calculates the average value for each windowing stride, and max-pooling calculates the maximum value for each stride. The pooling layer creates a downsampled feature map which is summarized from the input. This mechanism is helpful because small changes in the feature's location in the feedback detected could be detected in a pooled feature map with the feature in the exact location. The pooling layer formula is stated in equation (3) [2].

$$I_{i,j,k}^{t} = f_{0<x\leq d,0<y\leq d}(\mathrm{C}_{(i-1)^{*}s+x,(j-1)^{*}s+y,k}^{t}) \quad (3)$$

I is the input of this layer. F is the convolutional core. B is the bias.

B. CNN Architecture

The proposed custom CNN architecture consists of 18 hidden layers. The CNN model was built using the Keras library in TensorFlow implementation using Python [14]. The Keras sequential model is appropriate for a plain stack of layers with one input tensor and one output tensor. The first layer is a convolutional layer using 32 filters and kernel size 3x3. Fig. 1 shows the example K kernels that would be applied to the image during the convolutional process. The output of each convolution is an activation map. The first tuning process for the first layer is using bias_regularizer with L2 bias with 0.1. L2 is the sum of the square weights. The most common type of regularization is L2, also called "weight decay", with values often on a logarithmic scale between 0 and 0.1 [19].

The second layer was also a convolutional layer with different kernel sizes. Kernel size is smaller to filter different sizes of the image. The dimensionality reduction fits with the smaller image than the usual average image trained to the CNN. The third layer was a max-pooling layer. The results highlight the most present feature in the stride.

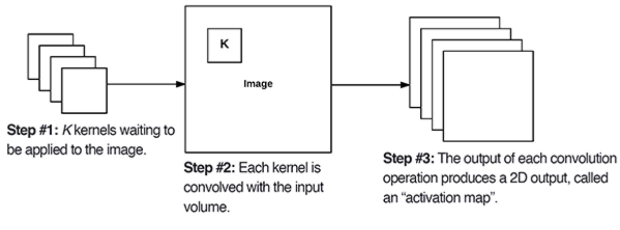

Fig. 1. Example of K kernels used in the convolutional process

This pooling has been better in practice than average pooling for computer vision tasks like image classification. The configuration of max-pooling used 2x2 pool size and 2x2 stride.

The fourth layer is the dropout layer. Dropout is a training strategy in which a subset of neurons is ignored and dropped out at random. The contribution of the activation of downstream neurons on the forward pass is removed temporally. Any weight updates are not applied to the neuron on the backward pass. As a result, the network becomes less sensitive to the weights of individual neurons, and the neural network will be able to generalize better. It will be less prone to overfit the training data. The dropout rate is set to 10%, meaning that one out of every ten inputs will randomly be omitted from each update cycle. The complete model architecture is shown in Table 1. Table 1 was the best model of face recognition tested with the student dataset. Architecture has 18 layers deep with a total of 117 million parameters. The input size of this architecture uses a default pixel used on typical CNN, which is 224x224 with three color channels.

The ninth layer until the twelfth layer is also the repetition from the previous layer. There is a difference in the number of filters which is multiplied by two and becomes 128. The design of the filters is gaining more detail in the deeper layer. Kernel size is also reduced by 2x2 to become 6x6 which still has a factor of 3.The padding is set to the "same" in the third section. Padding "same" tries to pad evenly left and right, but if the amount of columns to be added is odd, it will add the extra column to the right. When the kernel size is even, there is an advantage of zero padding. Zero padding means that there is no cut of the face on the left and right sides. There is an addition of the kernel regularizer to apply a penalty on the layer's kernel. The smaller output pixel with the deeper complex filter needs a reduction of overfitting in bias and weight because there have been many transformations before. The dropout rate was increased by 20% to make a more extensive weighted network that could forget some detailed features to decrease overfitting. In the last convolutional layer, filter size increased by about 256, and the kernel size used was the optimum size of 3x3. Bias and kernel regularizer is decreased to 0.0000001 because it is near 0.000001, which is all models used as the standard configuration [20]. The filter is going to have more detail, so the regularizer is tried to be decreased. The last convolutional layer is combined with a more extensive dropout layer which is 0.3. The reason is that there is trivial to make a weighted network forget some overfitted features to understand more about facial features. Combining this structured pattern is the best mechanism.

The next hidden layer is flattened. The flatten operator unrolls the values of the output shape to become one dimension. After the flattening, there is a fully connected layer known as the dense layer in the Keras module on Tensorflow. The advantage of a dense layer is that a thick layer offers learned features from all combinational elements of the previous layer. The dropout layer is added to a model between existing layers and applies to outputs of the last layer fed to the subsequent layer, which is also dense. This scenario makes a neuron from the previous fully connected layer set to zero with the probability of 20% according to the dropout rate. The final layer is dense with 16 units referring to the class of the different students. The activation function of softmax is applied to classify the nearest feature between the testing subject and the trained subject. Softmax activation identified the student as someone between the 16 classes, although the student is not included in the training class category.

TABLE I. THE ARCHITECTURE OF THE CUSTOM CNN WITH TENSORFLOW ENGINE

Layer (type)	Output Shape	Filter (Activation)	# Param	Kernel Size	Pool Size	Bias Regularizer	Kernel Regularizer	Dropout Rate
conv2d	210x210x32	32 (relu)	21632	15x15		0.1		
conv2d	198x198x32	32 (relu)	173088	13x13		0.1		
max_pooling	99x99x32		0		2x2			
dropout	99x99x32		0					0.1
conv2d	92x92x64	64 (relu)	131136	8x8		0.1		
conv2d	85x85x64	64 (relu)	262208	8x8		0.1		
max_pooling	42x42x64		0		2x2			
dropout	42x42x64		0					0.1
conv2d	42x42x128	128 (relu)	295040	6x6		0.1	0.1	
conv2d	42x42x128	128 (relu)	589952	6x6		0.1	0.1	
max_pooling	21x21x128		0		2x2			
dropout	21x21x128		0					0.2
conv2d	21x21x256	256 (relu)	295168	3x3		0.0000001	0.0000001	
dropout	21x21x256		0					0.3
flatten	112896		0					
dense	1024		115606528					
dropout	1024		0					0.2
dense	16		16400					

IV. EXPERIMENTS AND RESULTS

Many factors affect face recognition accuracies, such as the lighting, camera angle, and camera quality that require the quality of the GPU used in training. In this research paper, we propose a custom model of CNN because the complex existing could not resolve the face detection situation in this dataset. This section explains the experiments arranged and the results.

This research uses 3840 images of the students using 16 class of the students. One class of the student contains a variation of 240 images. The cross-validation uses the proportion of 70 percent data for training and 30 percent data for testing. The testing mechanism uses different photos from the cross-validation methods using the real-time CCTV camera. The testing mechanism uses 2500 images which are size is almost 70 percent of the total training photos. Fig. 2 shows the first layer of the convolutional layer and the second layer of the convolutional layer. In the first part, an example of the filter captured in the first layer goes deep and detailed in the next layer. These two figures explain that in the convolutional network, could exist different features according to some factors like face fixture and part exists in the facial image.

The max-pooling layer captures the aggregation of the two convolutional layers shown in Fig. 3. The max-pooling layer could represent some important features mixed. In the dropout layer, the feature captured is nearly the same as the max-pooling layer. In the dropout layer, the effect of the probability of the neuron weight is set to be zero, still not modifying the emergence feature captured in the previous layer. This mechanism will affect the accumulative weighing if the same feature also comes from another face. Fig. 5 shows the learning model of the accuracy in each epoch. Accuracy is still in line with the loss model that also decreased logarithmically. Although the training learning curve is near the testing learning curve, the results of testing using images that are not included in the training section are slightly different. The average accuracy of each class of the student goes down to 88.23%.

Fig. 2. The Keras Tensoflow Conv2D first and the second layer filtered feature

Fig. 3. The Keras Tensorflow Max Pooling and Dropout layer

There is three class of the person that accuracy of detection is lower than the other 12 class. Table 2 shows the accuracy of detection between 16 students and the average accuracy.

Fig. 6 shows the example of face recognition result from the image captured from the CCTV. The compilation of one class of the student is shown in Fig. 7.

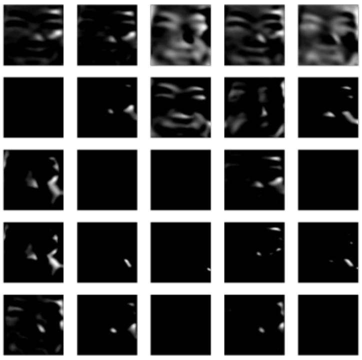

Fig. 4. The Keras Tensoflow Conv2D first layer filtered feature

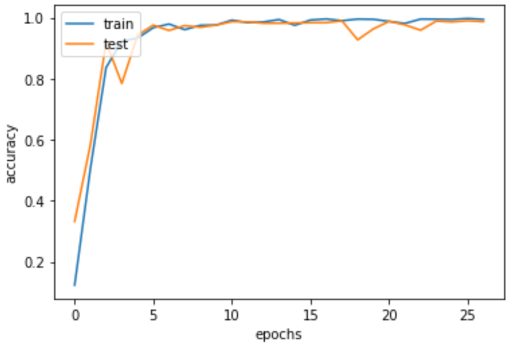

Fig. 5. The accuracy model of the training versus test state

TABLE II. ACCURACY OF EACH CLASS OF THE STUDENT AND THE AVERAGE ACCURACY

Person	**Accuracy (%)**	**Person**	**Accuracy (%)**
Person 1	92	Person 9	100
Person 2	84.61	Person 10	100
Person 3	65.11	Person 11	97.91
Person 4	57.84	Person 12	97.91
Person 5	37.57	Person 13	100
Person 6	82.94	Person 14	100
Person 7	95.83	Person 15	100
Person 8	100	Person 16	100
Average Accuracy		**88.2325**	

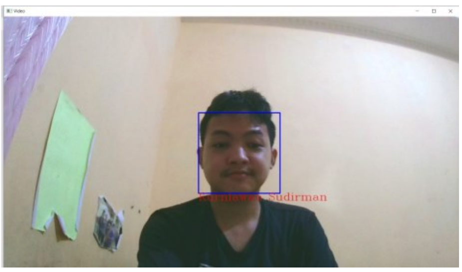

Fig. 6. Real time face recognition result

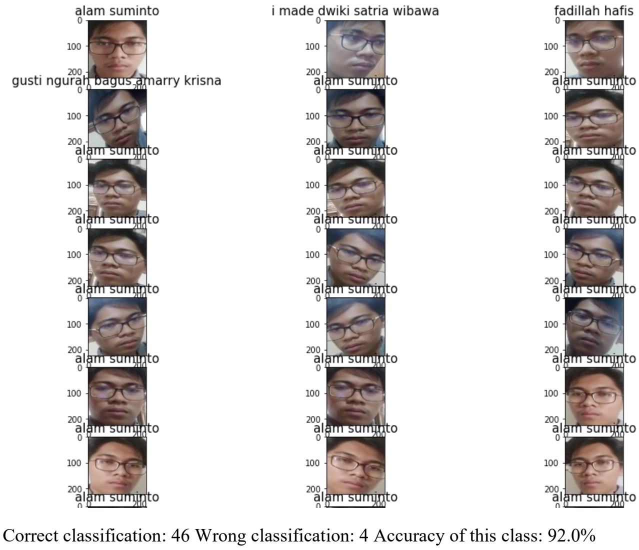

Fig. 7. Accuracy of the one class using the custom model

In the collection, there are some images shown with two labels that show the miss classification. In the MobileNetv2 transfer learning model tested, the accuracy is zero from the 48 photos of the example in Fig. 7 because all misclassified.

V. Conclusion

This research proposed a custom model of CNN based on the trial of making the proper layer based on the characteristic of each layer. This custom model effectively recognizes the face dataset of the student with a particular variation. The overall system's accuracy is better than the transfer learning model like MobileNetv2, which gives wrong accuracies in some classes. Although the deep of the layer is not very broad, the feature of the face could be captured. The custom model still uses the existing function and mechanism to create the hidden layer feature. The contribution is arranging the density of the layer and tuning with trial and error to get the best parametrization. The average accuracy from this model is 88% which needs to be improved in the attendance system. The attendance system for the future is required to make a delta calculation of the face detected based on the most detected class.

Acknowledgement

This work is supported and fully funded by Strategic Research Annual Budget Grand from the Faculty of Engineering Diponegoro University.

References

[1] M.-C. Huang, I.-P. Chen, H.-L. Huang, S.-F. Sung, and A.-G. Wang, "Project Design and Implementation of Face Recognition, Fever Detection, and Attendance Record Based on Sensing Technology," *Sensors and Materials*, vol. 33, no. 6, p. 1787, Jun. 2021, doi: 10.18494/SAM.2021.3203.

[2] W. Wang, J. Yang, J. Xiao, S. Li, and D. Zhou, "Face Recognition Based on Deep Learning," in *Human Centered Computing*, vol. 8944, Q. Zu, B. Hu, N. Gu, and S. Seng, Eds. Cham: Springer International Publishing, 2015, pp. 812–820. doi: 10.1007/978-3-319-15554-8_73.

[3] M. A. Shenbagharaman, B. Shunmugapriya, and M. K. Rajkumar, "Face Recognition Method For E-Attendance System," vol. 25, no. 4, p. 11, 2021.

[4] S. M. Bah and F. Ming, "An improved face recognition algorithm and its application in attendance management system," *Array*, vol. 5, p. 100014, Mar. 2020, doi: 10.1016/j.array.2019.100014.

[5] M. Gopila and D. Prasad, "Machine learning classifier model for attendance management system," in *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, Oct. 2020, pp. 1034–1039. doi: 10.1109/I-SMAC49090.2020.9243363.

[6] F. Z. Unal and M. S. Guzel, "A Comparison ff Deep Learning Based Architecture wth a Conventional Approach for Face Recognition Problem," p. 21.

[7] Q. Xiang, X. Wang, R. Li, G. Zhang, J. Lai, and Q. Hu, "Fruit Image Classification Based on MobileNetV2 with Transfer Learning Technique," in *Proceedings of the 3rd International Conference on Computer Science and Application Engineering - CSAE 2019*, Sanya, China, 2019, pp. 1–7. doi: 10.1145/3331453.3361658.

[8] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 815–823. doi: 10.1109/CVPR.2015.7298682.

[9] A. K. Dubey and V. Jain, "A review of face recognition methods using deep learning network," *Journal of Information and Optimization Sciences*, vol. 40, no. 2, pp. 547–558, Feb. 2019, doi: 10.1080/02522667.2019.1582875.

[10] S. Khan, A. Akram, and N. Usman, "Real Time Automatic Attendance System for Face Recognition Using Face API and OpenCV," *Wireless Pers Commun*, vol. 113, no. 1, pp. 469–480, Jul. 2020, doi: 10.1007/s11277-020-07224-2.

[11] A. S. Bist, "A Novel Approach for Facial Attendance:AttendXNet," *att*, vol. 2, no. 2, pp. 104–111, Jun. 2020, doi: 10.34306/att.v2i2.86.

[12] D. S. Trigueros, L. Meng, and M. Hartnett, "Face Recognition: From Traditional to Deep Learning Methods," *arXiv:1811.00116 [cs]*, Oct. 2018, Accessed: Sep. 19, 2021. [Online]. Available: http://arxiv.org/abs/1811.00116

[13] N. T. Son *et al.*, "Implementing CCTV-Based Attendance Taking Support System Using Deep Face Recognition: A Case Study at FPT Polytechnic College," *Symmetry*, vol. 12, no. 2, p. 307, Feb. 2020, doi: 10.3390/sym12020307.

[14] V. Koli, T. Vedak, and D. Sharma, "Object Detection based Attendance System," *International Journal of Engineering Research*, vol. 10, no. 04, p. 6.

[15] H. Patil, H. Khan, S. Ansari, and S. Sahu, "Automatic Attendance System Using Face Recognition," *Communication and Technology*, vol. 4, no. 1, p. 9, 2021.

[16] P. Bhavani, C. Saraswathi, K. Soba, and R. Preethika, "Automated Attendance System and Voice Assistance using Face Recognition," p. 11, 2021.

[17] A. Abraham, M. Bapse, Y. Kalaria, and A. Usmani, "Face Recognition Based Attendance System," p. 6.

[18] H. Tripathi, A. K. Srivastava, and A. Bisht, "Smart Attendance Portal using Facial Recognition," vol. 20, no. 3, p. 11, 2021.

[19] H. Pranoto and O. Kusumawardani, "Real-time Triplet Loss Embedding Face Recognition for Authentication Student Attendance Records System Framework," *JOIV : Int. J. Inform. Visualization*, vol. 5, no. 2, May 2021, doi: 10.30630/joiv.5.2.480.

[20] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton, "Regularizing Neural Networks by Penalizing Confident Output Distributions," *arXiv:1701.06548 [cs]*, Jan. 2017, Accessed: Sep. 30, 2021. [Online]. Available: http://arxiv.org/abs/1701.06548