Vol 4 (2022) E-ISSN: 2682-860X

International Journal on Robotics, Automation and Sciences

MASKED FACE RECOGNITION ATTENDANCE SYSTEM USING A MODIFIED CONVOLUTIONAL NEURAL NETWORK

Jun Jie How, Shing Chiang Tan* and Kim Soon Liew

***Abstract* – In this paper, a masked face recognition based attendance system is developed by modifying a version of convolutional neural network (CNN). In this regard, a Support Vector Machine is integrated in the CNN to replace its original Softmax classifier to perform the task. The performance of the modified CNN in recognizing masked faces in a 5-fold cross validation was compared that of other CNNs. The experimental results show high effectiveness of the proposed CNN (i.e. 98.92%) in recognizing masked faces for recording attendance.**

Keywords— deep learning, convolutional neural network, support vector machine, masked face images

I. INTRODUCTION

The attendance system was introduced in the workplace for a very long time. It is used to track a particular person's attendance and record the arrival time, break time, and knock off time. The purpose of recording time is to calculate the total number of working hours and make payments for the employer based on the recorded working hours. An advantage of attendance system is that the system can be used to record the attendance while saving human resource at the same time. Besides that, the system also can reduce human error when recording the attendance and calculating the total number of working hours.

In the early days, the attendance records were taken manually using paper and pen. It was a very time consuming process to record all attendances in a company. To overcome this problem, a punch card attendance system was introduced. One of the most commonly used attendance systems around the world is the manual punch card attendance system [1]. This system requires an employee to punch a time card upon arriving at the company and punch the time card again before leaving working place. The total working hours and overtime hours are manually added on a master time sheet by the administration where the sheets are referred to calculate the salary for each employee based on the recorded working hours. Unfortunately, this system has a disadvantage, i.e., the time card shall be changed every month after all slots have been filled.

An enhanced punch card attendance system, which replaces the time card with Radio Frequency Identification (RFID) card, is introduced,. The RFID card attendance system works exactly as in the manual punch card attendance system that an employee is required to check in to work and check out before leaving work place. Unlike the punch card system, this system uses the RFID technology to record the employee attendance [2]. Each employee is given an RFID card that contains the particulars of an employee. When the employee places the RFID card close to the machine, his/her attendance will be recorded in the system.

*Corresponding author: sctan@mmu.edu.my

Jun Jie How is a final year student with the Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia (e-mail: junjiehow123@gmail.com).

Shing Chiang Tan is with the Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia (e-mail: sctan@mmu.edu.my).

Kim Soon Liew is an Information Technology Engineer with CSC STEEL SDN BHD, Kawasan Industrial Ayer Keroh, 75450 Melaka, Malaysia (e-mail: liewks@cscmalaysia.com).

International Journal on Robotics, Automation and Sciences (2022) 4:23-29
https://doi.org/10.33093/ijoras.2022.4.4
Manuscript received: 16 May 2022 | Accepted: 21 June 2022 | Published: 8 July 2022

Published by MMU PRESS. URL: http://journals.mmupress.com/ijoras

Another type of attendance system is the biometrics attendance system where biometrics technology is used to record the employee attendance. All employees shall provide their biometrics such as fingerprint into the system before the system is used to recognize them for signing in attendances. This system is easy to use where a finger is simply put on the sensor and the attendance is recorded. It has overcome the problem of manually inserting the data into the system if the employee has forgotten to bring the card to the company.

Nowadays, the face recognition technology has become the most famous biometrics verification around the world. It is widely used in many applications, such as system access verification, video monitor system, human computer interaction, network security, tracking employees or students' attendance and so on. The face recognition technology has become popular due to its fast-processing speed and convenience for use [3]. The recognition process often takes only a few seconds which can save a lot of time in verifying a user's identity. Besides that, the user only needs to look at the camera during the recognition process without performing any other actions. In view of these advantages, the face recognition system has been used in the attendance system to replace the older biometrics attendance system [4]. Before using this face recognition attendance system, an employee shall register for his/her identity by providing his/her face image; this image will be stored in a database. When the attendance system is in operation, the face image of an employee is captured at the entrance and this image is checked against the stored images using a face recognition module before identifying the employee and signing in attendance. There are several types of face recognition models and deep learning models that have been used to perform face recognition, which are listed in Table 1.

Since the Covid-19 is spread around the world, a contactless face recognition attendance system is a feasibly set up in a company to replace the existing punch card attendance system that requires a physical interaction with employees. This system could prevent human contact that is helpful in reducing the risk of Covid-19 infection. The purpose of the research in this paper is to modify a deep learning model in data classification so as to investigate its ability in recognizing masked faces before signing in attendances.

II. LITERATURE REVIEW

A. *Human Face Recognition Using Machine Learning*

Lung et al. [6] proposed a face recognition model by using K-Nearest Neighbours (KNN) and Support Vector Machine (SVM). The first process was to collect the dataset. Face images were cropped and resized to 140x140. An ellipse mask with black background was assigned to process these images. This dataset contained the images of 52 persons where the number of images per person was 100. The next step is to perform features extraction from these images by using a ResNet-34, and the results were encoded in "pickle" files output vectors and labels. After that, 50 images per person were fed to the KNN and SVM to perform training and another 50 images per person were used for as a test dataset. The experimental results showed that the KNN and SVM model performed with a 98% test accuracy, the time taken for performing face recognition are 1.26 seconds and 1.13 seconds respectively. Nasr et al. [7] proposed a face recognition system using a machine learning approach. This project consisted of 3 phrases which were face detection, feature extraction and classification. First, the faces were detected using the viola Jones algorithms from the face database after cropping down face regions. The next step was to perform the feature extraction using Bag of Feature (BOF) to extract important features of the images. Those data were passed into a multi-class SVM to perform the classification on face images by using an error-correcting output code framework. The result of this experiment showed that the accuracy for face recognition using multi-class SVM was 99.21%.

Table 1. Face Recognition Models

Type	Description
Haar Cascades [5]	It works like a simple CNN (Convolutional Neural Networks) which extracts image features and choose the best features using Adaboost.
Dlib Frontal Face Detector [5]	It uses the Histogram of Oriented Gradients (HOG) to do the features extraction and forward those features to a support vector machine (SVM).
Multi-task Cascaded Convolutional Networks [5]	It obtains the candidate windows and their bounding box regression vector by using a fully convolutional network. It uses an on-maximum suppression (NMS) to group the highly overlapped candidates. After that, these candidates will be passed to another CNN to reject the number of false positives and perform calibration of bounding boxes.
DNN Face Detector in OpenCV [5]	It is a Caffe model based on the Single Shot-Multibox Detector (SSD) and the backbone of this model is using the ResNet-10 architecture.

.

Chawda et al. [8] proposed an SVM for face recognition. First, a database was collected by using the camera and Haar Cascade was applied to detect the human face. When the human face was detected, the images were captured and saved into the database. These images were converted into grey scale images. After that, the features of all these images were extracted using a Principal Component Analysis (PCA) algorithm. During feature extraction process, each of the face image was normalized by subtracting the mean face and each of this normalized face was a unique feature which was also called as eigenface. Next, these eigenfaces were divided into training and testing sets. Features were learned from the training set using an SVM. The experiment showed that the number of eigenfaces could affect the accuracy of face recognition. If the number of eigenfaces increased, the accuracy of this model decreased. The accuracy of this model was 99% when the number of eigenfaces was 75 and 97% when the number of eigenfaces was 150.

Karthik and Manikandan [9] proposed a Relevance Vector Machine (RVM) classifier to solve the face recognition in a real time system. First, the face images were processed by using Viola Jones algorithms to perform face detection and capture the face in the camera frame. The Histogram of Oriented Gradient (HOG) method was applied on these images to perform features extraction. All images were resized into 128 x 128 pixels and then split into 4 x4 cells with a size of 32 x 32 pixels. After that, all images were passed to the RVM classifier to perform training and testing. In this project, three different RVM architectures were used, which were Half Against Half architecture (HAH), Hierarchical Tree (HT) architecture and One Against All (OAA) architecture. The accuracy results of OAA, HT, and OAA were 90%, 94.5%, and 97% respectively. When running the face recognition process in real-time, the accuracy results were reduced to 81.25%, 87.5%, and 87.5% respectively.

B. Human Face Recognition Using Deep Learning

Winarno et al. [10] proposed a deep learning approach by combining Convolutional Neural Network with Principal Component Analysis (CNN-PCA) to perform face recognition. Winarno et al. [10] also used the Haar Cascade to capture face images. After face images were collected, they were normalized by a cropping technique and then resized. They were also converted from colour images into greyscale images wherein the brightness and contrast of those images were adjusted to improve the quality of greyscale images. All images were fed to a Convolutional Neural Network to build a 2D image reconstruction model to 3D. Next, the vector shape and texture were combined to generate a correlation point on the new face image that had similarities with initial image. All these images were used to form a database for face recognition. Feature extraction on face images was performed using the PCA method. PCA was used to reduce the dimension of face image resolution. The last process was to perform classification by using the Mahalanobis distance method. This method was used to determine the similarity of the facial features between training and testing sets. The experimental results were between 90% - 98% by using the CNN-PCA method.

Qu et al. [11] proposed a deep learning recognition network by using a CNN. The face database used in this project was collected from Carnegia Mellon University. The face images of 20 persons were collected where 120 face images of each person were used for training and 50 face images of each person were assigned to a testing set. All face images had a size of 32 x 32. The face data were passed into a CNN network with 4 hidden layers to perform training. These hidden layers consisted of 2 convolutional layers and 2 pooling layers. After performing 100 iterations, the recognition error rate was reduced to 0.75% and the accuracy of the face recognition was 99.25%. Qu et al. [11] also presented the results of PCA and Local Binary Pattern (LBP) of which the accuracy results were 91.6% and below 60% respectively. The experimental results showed that CNN achieved the highest accuracy in the study.

Arsenovic et al. [12] proposed a CNN model to perform face classification in a face recognition attendance system. First, a database was collected by capturing the face images of a person at different positions. Since the datasets used in Arsenovic et al. [12] was small, a data augmentation technique was applied to the original images. The data augmentation technique was performed to increase the size of the dataset so that it could help to increase the accuracy of the proposed model. Arsenovic et al. [12] added noise to the original images and blurred the images to increase the number of images. Arsenovic et al. [12] also performed data augmentation using the Dlib library to locate the user's face's identity such as eyes, nose, mouth and so on. By locating the part on the user's face, some of the accessories were added to the user's face such as glasses to generate new images to include in the training set. After collecting the images, the CNN cascade was used to detect the user's face region in the camera frame. After that, a pre-trained FaceNet network was used to perform the training with the dataset. Next, an SVM was applied to perform classification. The result of this experiment showed that the overall accuracy of this face recognition system was 95.02%.

Lv et al. [13] proposed a CNN with Extend Local Binary Pattern (ELBP) and Deep Convolutional Generative Adversarial Network (DCGAN) for face recognition under different illumination conditions. The ELBP is a feature extraction method which performs computation in a short period of time. This ELBP is an improved version of the traditional LBP method. ELBP overcomes limitations of traditional LBP where the latter is not able to process images in different sizes and textures and is also not able to perform circular operation. Image data under different illumination conditions are available in a small amount. The DCGAN is proposed to create new data based on original images in a small dataset. The intention of using DCGAN to increase the accuracy of face recognition. The D (i.e., discriminator) in the DCGAN represents a 4 layered

CNN for which each layer is activated by the Relu activation function. The G (i.e., generator) in DCGAN represents another CNN of 4 convolutional layers wherein the normalization process and Relu activation function adjustment are performed. All face colour images were available in the same size (i.e., 64 x 64 pixels) and converted into grey scale images. The ELBP was applied on these images to perform feature extraction. After that, all images were fed into the DCGAN to perform data generation and training. The test accuracy of DCGAN was 85%.

III. PROPOSED METHOD

A CNN consists of two parts, which are the convolutional layers and fully connected layers. The convolutional layers are used to extract important features from the images whereas the fully connected layers are used to perform the classification based on the extracted features. In this project, three variants of CNN (namely, VGG16, VGG19, and MobileNetV2) are utilized. The face recognition process is performed by imposing transfer learning in VGG16, VGG19 and MobileNetV2. In this section, explanation for transfer learning, data augmentation, and the architecture of VGG16, VGG19 and MobileNetV2 will be provided.

Transfer learning involves applying the knowledge learned by a machine/deep learning method from data in a source domain to solve a problem in a target domain [14]. Provided the tasks in the source and the target domains are closely related, in transfer learning, the model can use the knowledge learned from the previous task, and this knowledge is updated with information from a database of the new task. In this case, the model can learn the new knowledge faster. The widely used source domain in the transfer learning method is applied in computer vision is ImageNet, which is a huge dataset that contains a lot of images defined in many categories. The ImageNet consists of more than 14 million images that belong to 1000 classes. This ImageNet had been used to perform the transfer learning in many CNN models such as VGG16, VGG19, MobileNetV2, ResNet50, DenseNet121 and many other deep learning models. These pre-trained deep learning models can be applied to solve a different but related classification problem. These pre-trained models will apply the knowledge learned from the ImageNet to solve a new classification problem by updating them with the features from a new dataset. In a conventional machine/deep learning process, the model is built after learning knowledge of a task directly from a database; it is then applied to perform classification on provision of data from the same domain. Occasionally, pre-trained deep models are more efficient than their conventional counterparts in performing classification tasks.

The VGG16 was introduced by Karen Simonyan and Andrew Zisserman from University of Oxford in 2014 [15]. It is a deep learning model used to perform classification on ImageNet. Experimental results showed that VGG16 can achieve 92.7% test accuracy with the ImageNet dataset. This deep learning model is composed of 13 convolutional layers, 3 fully connected layers, 5 max pooling layers and it performs image data classification by using a SoftMax classifier. When colour images (in RGB) in a size of 224x224x3 are available, they are processed through 2 convolutional layers, 1 max pooling layer, 2 convolutional layers and again followed by 1 max pooling layer in VGG16. After that, the data is processed in 3 convolutional layers and followed by 1 max pooling layer of VGG16 for 3 times before being processed at 3 fully connected layers in which the first 2 fully-connected layers contain 4096 channels, and the last layer is a SoftMax that contains 1000 channels to classify those images in the ImageNet dataset. The VGG16 architecture is shown Figure 1.

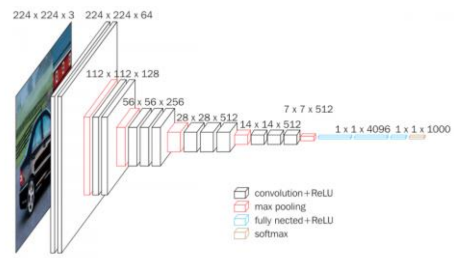

Figure 1. VGG16 Architecture [15]

VGG19 is a variant of the VGG model which has 16 convolutional layers, 5 max pooling layers, 3 fully connected layer, and lastly a SoftMax classifier [15]. Similar to VGG16, the input size of an image required to pass to the convolutional layers of VGG19 is 224x224x3. These images go through 2 convolutional layers and then 1 max pooling layer. The images are processed in another 2 convolutional layers and followed by a max pooling layer in one pass, and subsequently in 4 convolutional layers and 1 max pooling layer for 3 times. The features from the last max pooling layers are sent to 3 fully connected layers where the last fully connected layer is a SoftMax. The VGG19 architecture is shown in Figure 2.

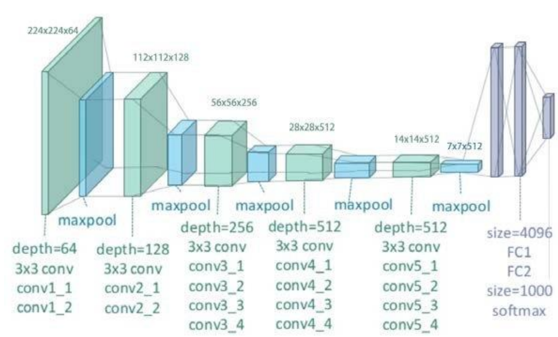

Figure 2. VGG19 Architecture [15]

MobileNetV2 is an improved version of deep neural network from MobileNet [16]. This CNN consists of 53 layers. It is designed for use in mobile devices that consumes less computing resources but still can perform with good classification performance. MobileNetV2 uses depthwise separable convolutions to replace the fully convolution operator. Depthwise separable convolutions perform convolutions into 2 layers. The first layer is named as depthwise convolution that conducts lightweight filtering on a single input data channel. The second layer is called pointwise convolution that performs 1 x 1 convolution. The main function of this layer is to construct a new feature based on input channel properties.

A standard convolution operator takes an input tensor of size $h_i \times w_i \times d_i$ and implements a convolutional kernel $K \in R^{k \times k \times d_i \times d_j}$ to generate an $h_i \times w_i \times d_i$ output tensor. The computational cost of this convolution layer is $h_i . w_i . d_i . d_j . k . k$. In MobileNetV2, the depthwise separable convolution performs a regular convolution operation at a lower cost, $h_i . w_i . d_i (k^2 + d_j)$. The computational cost of 3 x 3 depthwise separable convolution in MobilenetV2 is reduced around 1/8 or 1/9 from the cost of the standard convolution is applied. The accuracy of MobileNetV2 decreases slightly from that of MobileNet.

MobileNetV2 also uses the inverted residuals due to the usage of depthwise separable convolution. To gain extra features, the inverted residuals module's input will increase the channel through $i \times i$ convolution. After that, the 3×3 convolution is used to extract features and $i \times i$ pointwise convolution is applied to compress channel number.

SVM is a machine learning algorithm that is used for handling classification and regression problems [17]. In many circumstances, SVM is used to solve the classification problem. It will find out the hyperplanes in an *N*-dimensional space (where *N* represents the total number of features) that can best classify data. The hyperplanes in the SVM are the decision boundaries which are referred to classify data points. The main purpose of SVM is to maximize the margin between data points and the hyperplane. Hinge loss is a loss function that is used to maximize the margin. The formula of calculating the Hinge loss is shown as follows:

$$c = \begin{cases} 0, & if\ y * f(x) \geq 1 \\ 1 - y * f(x), & else \end{cases} \tag{1}$$

When the predicted value *f*(*x*) and the actual value *y* of a data point *x* have same sign, the cost *c* is zero. If this is not the case, then SVM will need to determine the loss value. To make a balance between margin maximization and loss, a regularization parameter λ is added into the loss function. The loss function is defined as follows:

$$min_w \lambda \parallel w \parallel^2 + \sum_{i=1}^{n} (1 - y_i(x_i, w)) \tag{2}$$

With loss function above, it uses the derivative with respect to the weight *w* to carry out gradients which are be used to update the weights. The formula of this gradient is shown as below:

$$\frac{\delta}{\delta w_k} \lambda \parallel w \parallel^2 = 2\lambda w_k$$

$$\frac{\delta}{\delta w_k} (1 - y_i(x_i, w))_+ = \begin{cases} 0, & if\ y_i(x_i, w) \geq 1 \\ -y_i x_{ik}, & else \end{cases} \tag{3}$$

If no misclassification occurs, the weight *w* will be revised by updating its existing value with the gradient from the regularization parameter only, as follows:

$$\omega = \omega - \alpha \cdot (2\lambda\omega) \tag{4}$$

If the misclassification problem occurs, the existing weight will be updated with a term that combines the loss and the regularization parameter. The weight update when misclassification occurs is shown as follows:

$$\omega = \omega + \alpha \cdot (y_i \cdot x_i - 2\lambda\omega) \tag{5}$$

The proposed CNN models (i.e, VGG16, VGG19 and MobileNetV2) are separately applied as a feature extractor whereas SVM is applied as a classifier to replace the Softmax to perform masked face recognition. The implementation of the proposed model is as follows:

1. Masked face image data are retrieved from the datasets and they are pre-processed.
2. The dataset is split into 2 subsets, which are a training set and a test set.
3. VGG16, VGG19 and MobileNetV2 are used to extract the features from the training dataset.
4. The extracted features are passed to the SVM classifier to perform feature learning (or training).
5. The performance of the proposed model is evaluated by using the test set.

In this research work, three deep learning models (i.e., VGG16, VGG19, and MobileNetV2) adopting transfer learning are applied to classify user's face with mask. By using the pre-trained weights, these deep learning models can perform feature extraction from the face masked dataset and an SVM classifier is applied to perform feature learning and then image classification.

IV. Experimental Results

A. Dataset

A face dataset consisting of 330 face images of 11 persons who wore mask were collected. The image of a person who wore a mask was captured from 3 directions, which were from front, left and right. To collect an adequate number of images, the image of each person who wore a mask at a direction was captured for 10 times. This dataset is organized in 11 folders of facemask image data where each folder represents one person. Each folder was labelled with a number from 0 to 10.

Figure 3. Image Samples

B. Data Preprocessing and Data Augmentation

In this experiment, the computing procedure is divided into 3 parts, which are data augmentation, pre-processing, and deep learning with VGG16, VGG19, or MobileNetV2. All images are pre-processed and then split into a training set and a test set. VGG16, VGG19, or MobileNetV2 is used to learn knowledge from the dataset through a transfer learning method. A test set is used to evaluate the classification performance of the trained model.

Before performing data pre-processing, a data augmentation technique is applied on the dataset by flipping each image to the right side and increase the contrast of each image to increase the total number of images. The contrast of each image is also adjusted. All images generated by the data augmentation technique are saved in the same folder. All images in a folder will be converted into the RGB format in a size of 96x96x3. All images are shuffled before being divided into a training set and a test set in a 70:30 ratio. The next step is data normalization in which the pixel values of all images from the training and test sets are normalized into a value within a range between 0 and 1 by dividing each pixel value by 255.

C. Results and Analysis

In this experiment, the VGG16, VGG19 and MobileNetV2 are called from the Keras Application and the pre-trained weights from the ImageNet are used to perform transfer learning. The size of image data is reduced from 224x224x3 to 96x96x3; and these image data are used for training with the VGG16, VGG19 and MobileNetV2. The convolutional layer of these three deep learning models is kept unchanged but the last fully connected layer of each model, which is a Softmax, is replaced with SVM to perform feature learning and data classification. In this work, the SVM kernel is set to a linear function.

To evaluate the performance of the purpose model, a 5-fold validation technique had been applied to evaluate the performance. The results had been shown in Table 2.

Table 2. 5-Fold Average Testing Accuracy Result between VGG16, VGG19, MobileNetV2, VGG16 + SVM, VGG19 + SVM, and MobileNetV2 + SVM

Deep Learning Model	Average Testing Accuracy Result
VGG16	97.37%
VGG19	96.59%
MobileNetV2	98.25%
VGG16 + SVM	81.75%
VGG19 + SVM	83.16%
MobileNetV2 + SVM	98.92%

The results from Table 2 show that the MobileNevtV2 + SVM model has achieved the highest test accuracy (98.92%) among other deep learning models. In other words, the MobileNetV2 + SVM model can achieve a very high accuracy performance in recognizing masked faces.

V. Conclusion

This research work is aimed to investigate the effectiveness of three deep learning models (VGG16, VGG19 and MobileNetV2) that are each equipped with an SVM for recognizing mask faces before confirming individuals' attendance. The results show high accuracy and thus, demonstrates the usefulness of MobileNet-V2 + SVM in recognizing masked face images of different individuals. To further improve accuracy in face recognition, we intend to develop an ensemble version of deep learning models for performing masked/unmasked face recognition in the future.

References

[1] A. Fong, "Fingerprint: Card access control and time attendance solutions : Fingertec Worldwide," *Fingerprint | Card Access Control And Time Attendance Solutions : FingerTec Worldwide*, Jun-2005. [Online]. Available: https://www.fingertec.com/.

[2] C. B. Chew, M. Mahinderiit-Singh, K. C. Wei, T. W. Sheng, M. H. Husin, N. Hashimah, and A. H. Malim, "Sensors-enabled smart attendance systems using NFC and RFID Technologies," *International Journal of New Computer Architectures and their Applications*, vol. 5, no. 1, pp. 19–28, 2015.

[3] N. Kar, M. K. Debbarma, A. Saha, and D. R. Pal, "Study of implementing automated attendance system using face ... - IJCCE," *International Journal of Computer and Communication Engineering*, vol. 1, no.2, pp. 100-103, 2012.

[4] S.-H. Lin. (2000). "An Introduction to face recognition technology," *Informing Science*: *The International Journal of an Emerging Transdicipline*, vol. 3, no 1, pp. 001 – 007, 2000.

[5] V. Agarwal, "Face detection models: Which to use and why?," *Medium*, 02-Jul-2020. [Online]. Available: https://towardsdatascience.com/face-detection-models-which-to-use-and-why-d263e82c302c.

[6] L. F. Lung, M. N. Barua, and P. S. Juarez, "An image acquisition method for face recognition and implementation of an automatic attendance system for events," in *2019 IEEE XXVI International*

Conference on Electronics, Electrical Engineering and Computing (INTERCON), 2019, pp. 1–4.

[7] S. Nasr, K. Bouallegue, M. Shoaib, and H. Mekki, “Face recognition system using bag of features and multi-class svm for robot applications,” in *2017 International Conference on Control, Automation and Diagnosis (ICCAD)*, 2017, pp. 263-268.

[8] V. Chawda, V. Arya, S. Pandey, Shristi, and M. Valleti, “Unique Face Identification System using machine learning,” 2020 Second International Conference on Inventive Research in Computing Applications (ICIRICA), 2020, pp. 701-706.

[9] H. S. Karthink and J. Manikandan, “Evaluation of relevance vector machine classifier for a real-time face recognition system,” in *2017 IEEE International Coference on Consumer Electronics-Asia (ICCE-Asia)*, 2017, pp. 26-30.

[10] E. Winarno, I. H. A. I. Amin, H. Februariyanti, P. W. Adi, W. Hadikurniawati, and M. T. Anwar, “Attendance system based on face recognition system using cnn-pca method and real-time camera,” in *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2019, pp. 301-304.

[11] X. Qu, T. Wei, C. Peng, and P. Du, “A fast face recognition system based on deep learning,” in *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, 2018, pp. 289-292.

[12] M. Arsenovic, S. Sladojevic, A. Anderia, and D. Stefanovic, “Facetime - deep learning based face recognition attendance system,” in *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*, 2017, pp. 000053-000058.

[13] T. Lv, C. Wen, J. Zhang, and Y. Chen, “A face recognition algorithm based on cnn with elbp and dcgan,” in *2020 International Symposium on Computer Engineering and Intelligent Communications (ISCEIC)*, 2020, pp. 99-102.

[14] S. Chen, W. Liu, and G. Zhang, “Efficient transfer learning combined skip-connected structure for masked face poses classification,” *IEEE Access*, vol. 8, pp. 209688-209698, 2020.

[15] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv: 1409.1556, 2014. [Online]. Available: https://arxiv.org/pdf/1409.1556.pdf.

[16] S. Mark, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” arXiv preprint arXiv: 1801.04381v4, 2019. [Online]. Available: https://arxiv.org/pdf/1801.04381.pdf.

[17] L. Chen, “Support Vector Machine - simply explained,” *Medium*, 07-Jan-2019. [Online]. Available: https://towardsdatascience.com/support-vector-machine-simply-explained-fee28eba5496.