

# Web-based University Classroom Attendance System Based on Deep Learning Face Recognition

Nor Azman Ismail<sup>1</sup>, Cheah Wen Chai<sup>1</sup>, Hussein Samma<sup>1</sup>, Md Sah Salam<sup>1</sup>, Layla Hasan<sup>1</sup>,  
Nur Haliza Abdul Wahab<sup>1</sup>, Farhan Mohamed<sup>1</sup>, Wong Yee Leng<sup>1</sup>, and Mohd Foad Rohani<sup>1</sup>

<sup>1</sup>School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia,  
Johor Bahru, Skudai 86000 Malaysia

[e-mail: azman@utm.my, cheahwen1997@graduate.utm.my, hussein.samma@utm.my, sah@utm.my,  
l.hasan2@yahoo.co.uk, nur.haliza@utm.my, farhan@utm.my, wong@utm.my, foad@utm.my]

\*Corresponding author: Nor Azman Ismail

*Received September 27, 2021; revised December 5, 2021; accepted February 2, 2022;  
published February 28, 2022*

---

## Abstract

Nowadays, many attendance applications utilise biometric techniques such as the face, fingerprint, and iris recognition. Biometrics has become ubiquitous in many sectors. Due to the advancement of deep learning algorithms, the accuracy rate of biometric techniques has been improved tremendously. This paper proposes a web-based attendance system that adopts facial recognition using open-source deep learning pre-trained models. Face recognition procedural steps using web technology and database were explained. The methodology used the required pre-trained weight files embedded in the procedure of face recognition. The face recognition method includes two important processes: registration of face datasets and face matching. The extracted feature vectors were implemented and stored in an online database to create a more dynamic face recognition process. Finally, user testing was conducted, whereby users were asked to perform a series of biometric verification. The testing consists of facial scans from the front, right (30 – 45 degrees) and left (30 – 45 degrees). Reported face recognition results showed an accuracy of 92% with a precision of 100% and recall of 90%.

---

**Keywords:** Deep Learning, Pre-trained Model, Registration of Face Datasets, Face Recognition, Web-based Attendance System, Feature Vectors.

## 1. Introduction

Class attendance is very important as higher education institutions require the lecturer to record and monitor student attendance. Unfortunately, recording the attendance instantly and effectively becomes a very critical challenge. There is a belief that student attendance can predict student achievement. The traditional way of recording attendance is irrelevant, and technology like image processing and biological attendance records are more convenient.

Identity verification and recognition using biological characteristics have become more ubiquitous in the digital era because of convenience and concealment [1]. Traditional verification methods, which include access cards, work cards, and ID cards, faced difficulty adapting to the rapid pursuit of the current society. These methods have resulted in unwanted congestion, wastage of human resources, and time delay. Biometrics techniques have developed personal identification technology for access privileges and information systems [2]. Over the past few decades, face recognition has been the most imperative application among biometric-based systems in the field of computer vision [3, 4].

There are metrics such as knowledge-based, physiological biometrics-based, behavioural biometrics-based, and two or multi-factor information authentication in user authentication [5]. Knowledge-based authentication is the approach where identity-related information is secret to the users, which can be in digit PINs or lock patterns. Physiological biometric-based authentication utilises human body components such as fingerprints, hand geometry, face, and iris pattern. Behavioural biometric-based authentication tracks the users' behaviour, such as voice, gait, handwriting, etc. Two or multi-factor information authentication merge two or more metrics, such as a fingerprint with face and RFID with the face [5].

Face recognition is a physiological biometric technique. The process of facial recognition includes image acquisition, face detection, feature extraction, face recognition, and finally, verification or identification. The techniques of feature extraction in face recognition such as Principal Component Analysis (PCA) [6], Linear Discriminant Analysis (LDA), Local Binary Patterns (LBP) [7], Elastic Bunch Graph Matching (EBGM) [8], Gabor Wavelet [9], and Convolutional Neural Networks (CNN) [10].

CNN is a Deep Learning (DL) algorithm which is a class of Artificial Neural Network (ANN). The CNN can take an input image, learnable weights and biases to various aspects in the image, and it also has the ability to differentiate one from the other. The ability to extract detailed information from input images make DL be the best image processing method.

DeepFace [10] and FaceNet [11] utilise the CNN technique tremendously to make a huge breakthrough in facial recognition under unconstrained environments such as pose variation, face occlusion, poor illumination. The accuracy rate of DeepFace is 97.25%, whereas the accuracy rate of FaceNet is 99.63% using the LFW benchmark dataset.

Current face recognition systems appear to perform well in relatively controlled conditions but fail in some factors such as variations in posture, occlusion, lighting, low-resolution, ageing, and make-up [12]. Therefore, an improvement is needed to tackle the problem, especially in variations in posture scenarios. The difficulties of the face recognition problem can be derived from two aspects: large concentration on the frontal face image size and the database images [12].

The aim of this paper is to propose a web-based attendance system for university classrooms with the integration of deep learning face recognition pre-trained models. The

face recognition performance is measured based on the pose variation in the production site, which involves the participation of the real users. The proposed attendance system is dynamic with the integration of the database to store all the facial features data from each student.

The rest of the paper is organised as follows. Section 2 overviews related works on the comparative study of face recognition techniques and the attendance applications. Section 3 provides a comprehensive description of the proposed attendance system. Section 4 presents the results and discussions of the proposed system. Finally, the paper is concluded in Section 5.

## 2. Related Work

The development of face recognition systems focuses on four stages of approaches from the oldest to the latest: a statistical approach, holistic approach, machine learning, and deep learning [12]. Statistically based approaches used Eigenvectors, Eigenfaces, Hidden Markov Model (HMM) and so on. Holistic approaches focused on the whole face area, such as the mixture of PCA and LDA, spectroface and active appearance model (AAM). The evolution of 3D face recognition focused on machine learning approaches involving scale-invariant feature transform (SIFT), Face-Specific Subspace (FSS), Generalized Multiview Analysis (GMA) and so on. Recently deep learning approaches focused more on recognition using the deep learning approach such as Deep Neural Network (DNN), Discriminant Deep Metric Learning (DDML), Deep Correlation Feature Learning, Deep Canonical Correlation Analysis, Deeply Coupled Auto-encoder Networks (DCAN) and so on. **Table 1** summarises the most common face recognition techniques such as PCA, LDA, EBGM, and CNN.

**Table 1.** Face Recognition Techniques

Technique	Approach	Algorithm summary
PCA [6]	Global	Find a direction of lower-dimensional space of projected data with maximum total variance, which clusters the neighbours of the same class
LDA [12]	Global	Obtain a maximum ratio of between-class scatter matrix of samples to the within-class scatter matrix
EBGM [8]	Template	Locate and extract the facial features for face matching
CNN [10, 11]	Local	A representation learning approach by applying multiple filters to extract the features in a stack of layers.

**Table 2** shows the recent development of the attendance management system (from the year 2016 to the year 2019). It shows a bias toward IoT deployment and low use of the biometric system. Such attendance systems are Bluetooth low energy attendance android application [13], speech-based attendance system [14], bimodal biometric attendance system [15], location-aware event attendance system using QRCode and GPS technology [16] as well as IoT Based Smart Attendance System using RFID [17]. The common use of biometric-based systems may be due to the low accuracy rate and unreliable prediction under an uncontrolled environment. As for the IoT based attendance systems are convenient, fast, and reliable in any condition, thus giving a high percentage of use in the system.

**Table 2.** Attendance Management System Based On Different Types

Application/System	Technique	Limitation
Bluetooth Low Energy Attendance Android Application [13]	<ul style="list-style-type: none"> <li>- Use Bluetooth system for attendance marking</li> <li>- Show the performance of students (i.e., Day present, Total lecture)</li> <li>- Use an infrared motion analyser to count the number of students passing through the door into the classroom</li> </ul>	<ul style="list-style-type: none"> <li>- Difficult to sense the Bluetooth connection in a wide space.</li> </ul>
Speech Based Attendance System [14]	<ul style="list-style-type: none"> <li>- Use speech recognition for attendance markings.</li> <li>- Apply Power Spectral Density (PSD) and Transition Parameter for feature extraction.</li> </ul>	<ul style="list-style-type: none"> <li>- Low accuracy (roughly 60% only) due to the presence of noise and low instrument quality</li> </ul>
Bimodal Biometric Attendance System [15]	<ul style="list-style-type: none"> <li>- Combine fingerprint and face recognition for attendance marking</li> </ul>	<ul style="list-style-type: none"> <li>- Congestion problem due to dual authentication of an individual.</li> </ul>
Location-aware Event Attendance System using QR Code and GPS Technology [16]	<ul style="list-style-type: none"> <li>- Use QR code and GPS to track the attendance require a network to get the location of students.</li> </ul>	<ul style="list-style-type: none"> <li>- Angle for handling the device needs to be 90 degrees for QR Code scanning</li> <li>- Distance between the device and QR Code cannot be too far (below 9cm)</li> <li>- Appropriate illumination (above 60%) for QR Code</li> </ul>
IoT Based Smart Attendance System using RFID [17]	<ul style="list-style-type: none"> <li>- Use RFID for attendance marking.</li> <li>- Send a message to the parent about the attendance status of their children</li> </ul>	<ul style="list-style-type: none"> <li>- High cost of RFID reader</li> <li>- Card misuse or stolen</li> </ul>

### 3. Face Recognition Attendance System

This work used pre-trained models for face detection, facial landmark detection, and face features extraction. The detection models for face and facial landmark employs Single Shot Detector (SSD) network, where 68 landmark points were detected to be used in the face alignment process. Residual Network (ResNet) was used to get 128 values of face features within the region of interest of the detected face. These features were normalised and compared with user database features using Euclidian distance measure for label matching. The following sections explain the face recognition attendance system in detail.

#### 3.1 Choosing Required Deep Learning Face Recognition Models

face-api.js [18] is an open-sourced Javascript face recognition API using deep convolutional neural networks built using TensorflowJS as in Fig. 1. It provides some functions such as face detection, facial expression analysis, age and gender prediction, facial landmarks detection, and face recognition. There is a total of 18 files in the “weights” directory, as shown in Fig. 2. In the “weights” directory of the API, a combination of face detection, facial landmark detection, and feature extraction model is required to enable face recognition. For the face detection task, the required files are “ssd\_mobilenetv1\_model-shard1,” “ssd\_mobilenetv1\_model-shard2” and “ssd\_mobilenetv1\_model-

weights\_manifest.json,” For the facial landmark detection task, the required files are “face\_landmark\_68\_tiny\_model-shard1” and “face\_landmark\_68\_tiny\_model-weights\_manifest.json,” For feature extraction task, the required files are “face\_recognition\_model-shard1,” “face\_recognition\_model-shard2” and “face\_recognition\_model-weights\_manifest.json,” **Table 3** shows a brief description of the three required deep learning pre-trained models. Those model was based on existing models and integrated into the proposed system. The model is built on top of neural networks architecture as documented by the model creator. Euclidean Distance was measured between embedding without training process because it requires retraining when a new person is added to the datasets. For the final results, a series of detection, alignment and feature extraction techniques are applied to produce a list of vector values. Appropriate threshold values are selected to tweak the accuracy. Further sections specify the details of the process.

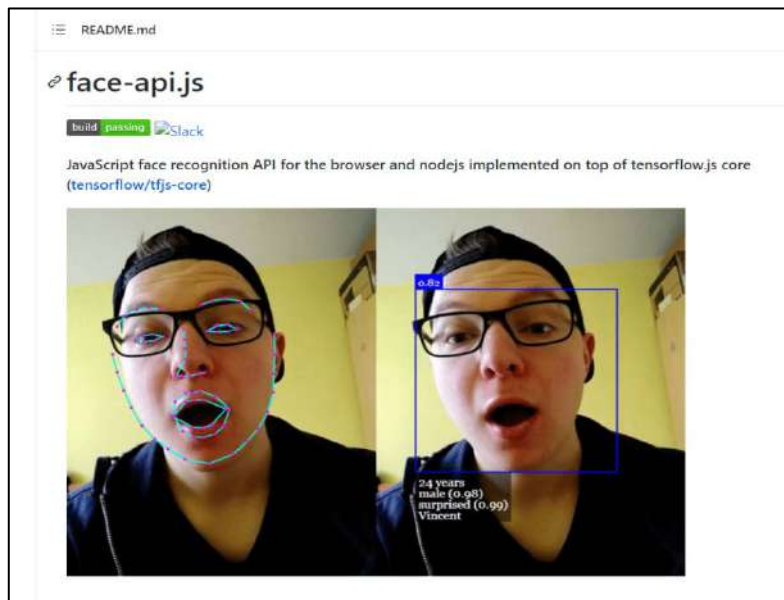


Fig. 1. face-api.js [18] on Github.

age_gender_model-shard1	check in age_gender_model + AgeGenderNet loading from weightmap	
age_gender_model-weights_manifest...	check in age_gender_model + AgeGenderNet loading from weightmap	
face_expression_model-shard1	add final expression recognition model + global api	
face_expression_model-weights_m...	add final expression recognition model + global api	
face_landmark_68_model-shard1	retrained landmark 68 and tiny landmark 68 point models	
face_landmark_68_model-weights_...	retrained landmark 68 and tiny landmark 68 point models	
face_landmark_68_tiny_model-shar...	retrained landmark 68 and tiny landmark 68 point models	Facial Landmark Detection Weights & Metadata
face_landmark_68_tiny_model-weig...	retrained landmark 68 and tiny landmark 68 point models	
face_recognition_model-shard1	recreate quantized shards of face recognition model, which apparently...	Feature Extraction Weights & Metadata
face_recognition_model-shard2	recreate quantized shards of face recognition model, which apparently...	
face_recognition_model-weights_m...	recreate quantized shards of face recognition model, which apparently...	
mtcnn_model-shard1	added mtcnn e2e tests + check in final model weights	
mtcnn_model-weights_manifest.json	added mtcnn e2e tests + check in final model weights	
ssd_mobilenetv1_model-shard1	only keep quantized models in this repo + renamed face_detection_mode...	Face Detection Weights & Metadata
ssd_mobilenetv1_model-shard2	only keep quantized models in this repo + renamed face_detection_mode...	
ssd_mobilenetv1_model-weights_...	only keep quantized models in this repo + renamed face_detection_mode...	
tiny_face_detector_model-shard1	check in latest tiny_face_detector_model	
tiny_face_detector_model-weights_...	check in latest tiny_face_detector_model	

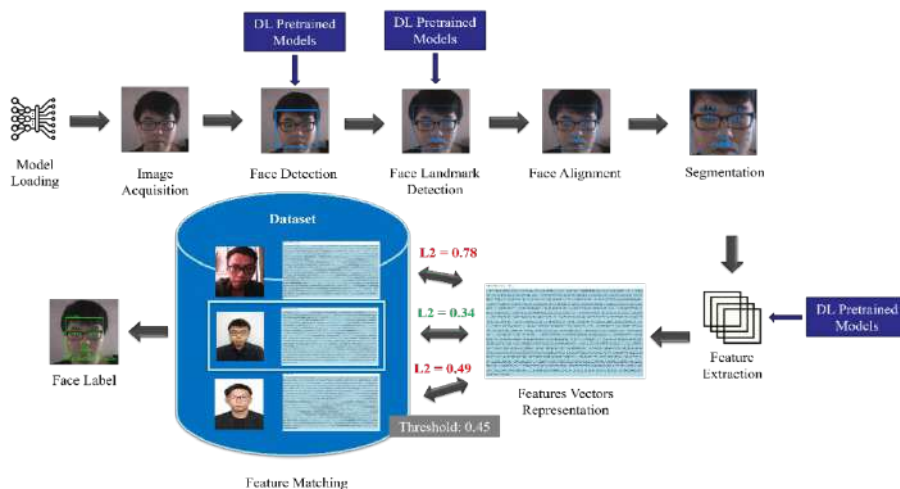
Fig. 2. Deep Learning Pre-trained Models Weights & Metadata in “weights” directory of API [18]

**Table 3.** Description of Chosen Deep Learning Pre-trained Models

Task	Deep Learning Algorithm of Pre-trained Models	Purpose
Face Detection	SSD MobileNet V1 [19]	Detect the location of faces and return the score, coordinates, width and height of the face area
Facial Landmark Detection	Depth-wise separable convolutions with densely connected block	Detect 68 landmark points and return the coordinates to be referenced for face alignment pre-preprocessing
Feature Extraction	ResNet-34 [20] like architecture	Extract 128 values of normalised feature vectors from the face ROI

### 3.2 Procedure of Face Recognition

First of all, all the weight files from the three pre-trained models need to be loaded into the system memory. Next, the image of BLOB type is fed into the face detection model with a 0.8 confidence score threshold to detect positive or negative class, followed by facial landmark detection and face alignment. The aligned result of the image is cropped and undergo feature extraction to generate a list of 128 values of feature vectors. A feature vector is a numerical representation of an object, which could be derived from the pixel intensity, edges and area of the face. Next, the feature vectors in the database are fetched to match with the detected feature vectors. A matching threshold is determined by the computed shortest Euclidean Distance value. The purpose of this threshold is to detect between the positive and negative classes. The nearest value of L2 to the 'zero' (0), the closest the recognised face label has to match, and vice versa. Other researchers, as in Fig.3, use L2 values of 0.78, 0.49, 0.34. To reduce the False Positive case (large L2 will cause mismatch), the value of threshold 0.45 is chosen to enable the unique scenario. Finally, the face ID from the datasets with the shortest difference of feature vector similarity using Euclidean Distance is the recognised face and mark with the label in the image canvas accordingly. The visualisation of the face recognition procedure is demonstrated in Fig. 3. The face recognition task is divided into two important processes: the registration of face datasets and face matching.

**Fig. 3.** Visualisation of Face Recognition Procedure.



### 3.3 Registration of Face Datasets

Each user with a student role is required to upload their face photograph for face registration. In the registration of face datasets, face detection is performed on the input image to detect the location of the face. If the confidence score of a detected face is beyond the minimum confidence score threshold, which is 0.8, a bounding box will be drawn onto the image canvas to indicate the face location. Next, facial landmark detection is performed to detect 68 key points for face alignment. The feature extraction is carried out to generate the feature vectors. If there are more than one face feature vectors generated which indicate more than one face is detected, the system will reject the registration to avoid matching error from the datasets during the evaluation. The extracted feature vectors are the 128 normalised values shown in [Fig. 4](#), combined into String datatype to be stored in the database. [Fig. 5](#) shows the pseudocode and workflow of face dataset registration. [Fig. 6](#) shows the feature vectors stored in the MongoDB database.

```
▼ descriptor: Float32Array(128)
  ▼ [0 ... 99]
    0: -0.03517751768231392
    1: 0.02568182907998562
    2: 0.072407566010952
    3: -0.009485913440585136
    4: -0.04103800281882286
    5: -0.07855777442455292
    6: -0.05235551297664642
    7: -0.10324186086654663
    8: 0.10022732615470886
    9: -0.0838657096028328
    10: 0.2091512233018875
    11: -0.10428614169359207
    12: -0.2487381547689438
    13: -0.11084699630737305
    14: -0.02561965398490429
    15: 0.1318848580121994
    16: -0.13264073431491852
    17: -0.09192939102649689
    18: -0.04829888790845871
    19: -0.047912657260894775
    20: 0.09070347249507904
    21: -0.000016939344277489
    22: 0.03301561251282692
    23: 0.06885295361280441
    24: -0.06539686769247055
    25: -0.3784627914428711
    26: -0.11294436454772949
    27: -0.1291082203388214
    28: 0.0065426304936409
    29: -0.04912027344107628
    30: -0.019528036937117577
    31: 0.06930074840784073
    32: -0.1553729772567749
    33: -0.032022349536418915
    34: -0.014265288598835468
    35: 0.0505460649728775
```

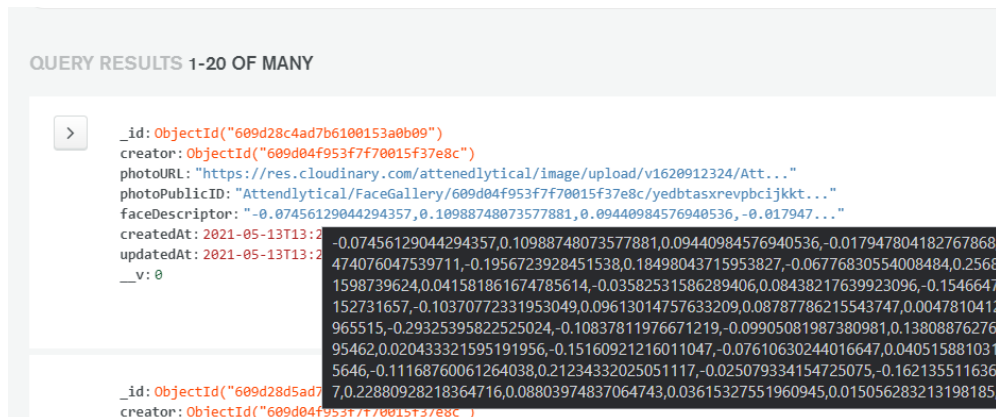
[Fig. 4](#). 128 values of normalised feature vectors.

```

1. start
2. initiate image acquisition
3. initiate face detection
4. IF Bounding Box Score is more than the minConfidence Score
   mark bounding boxes (detected face)
   ELSE
   go to step 2
5. initiate facial landmark detection
6. mark detected facial landmark points with dot
7. align detected bounding box
8. initiate feature extraction
9. get 128 values of Feature Vectors, converted to string format
10. IF single face only
    IF save photo
        record Feature Vectors string into database
    ELSE
        go to step 2
    ELSE
        go to step 2
11. database receives query
12. database processes query
13. database stores the Feature Vectors string
14. receive result (success or fail)
15. end

```

**Fig. 5.** Pseudocode of Registration of Face Datasets.



**Fig. 6.** Feature vector strings in MongoDB database.

### 3.4 Face Matching

The matching process is performed when taking the attendance. During the matching process, all the feature vectors of face datasets in the course are fetched from the server, and the results are stored into a JSON array as in [Fig. 7](#). The retrieved feature vectors string is parsed into Float32Array datatype. Next, the image capturing device is opened to allow for image acquisition. For every 0.2 seconds, the system will perform face detection to check for the current frame and check for the face's location as the same as the registration process of face datasets. After the feature extraction, the detected feature vectors are matched between the other feature vectors using Euclidean Distance. As demonstrated in [Fig. 8](#), the matching threshold, the shortest distance, indicates the recognised face. The respective face will be



marked with the student's name and matric number. The attendance transaction (Fig. 9) is stored in the database. Fig. 10 illustrates the pseudocode of the face matching process during attendance taking.

```

Array(20)
  0: {
    attend_at: "2021-05-22T01:46:21.904Z"
    FacePhotos: Array(3)
      0: {
        faceDescriptor: "-0.05484543368220329,-0.0004096481716260314,0.10484595596790314,0.02523457072675228,-0.04654...
        photoURL: "https://res.cloudinary.com/attenedlytical/image/upload/v162...
        __typename: "FacePhoto"
        [[Prototype]]: Object
      }
      1: {
        faceDescriptor: "-0.037855878472328186,-0.03608817234635353,0.08167783915996552,0.019783001393079758,-0.06548...
        photoURL: "https://res.cloudinary.com/attenedlytical/image/upload/v162...
        __typename: "FacePhoto"
        [[Prototype]]: Object
      }
      2: {
        faceDescriptor: "-0.05424008518457413,0.046237312257289886,0.074079_30633163,0.13684824109077454,-0.0342025...
        length: 2
        [[Prototype]]: Array(0)
      }
    student: {
      cardID: "A17CS0028"
      firstName: "Chai"
      lastName: "Cheah Wen"
      profilePictureURL: null
      __typename: "Person"
      id: "609d2b98ad7b6100153a0b1b"
      [[Prototype]]: Object
    }
    __typename: "FaceProfile"
  }
  1: {student: {...}, facePhotos: Array(2), __typename: "FaceProfile", attend_at: "2021-05-22T04:14:37.757Z"}
  2: {student: {...}, facePhotos: Array(2), __typename: "FaceProfile", attend_at: "2021-05-23T11:44:01.056Z"}
  3: {student: {...}, facePhotos: Array(2), __typename: "FaceProfile"}
  4: {student: {...}, facePhotos: Array(3), __typename: "FaceProfile", attend_at: "2021-05-22T05:08:01.059Z"}
  5: {student: {...}, facePhotos: Array(2), __typename: "FaceProfile", attend_at: "2021-05-22T07:43:28.825Z"}
  6: {student: {...}, facePhotos: Array(2), __typename: "FaceProfile"}
  7: {student: {...}, facePhotos: Array(2), __typename: "FaceProfile"}
  8: {student: {...}, facePhotos: Array(2), __typename: "FaceProfile", attend_at: "2021-05-24T04:41:46.515Z"}
  9: {student: {...}, facePhotos: Array(2), __typename: "FaceProfile", attend_at: "2021-06-04T06:11:15.161Z"}
  10: {student: {...}, facePhotos: Array(3), __typename: "FaceProfile", attend_at: "2021-05-22T07:07:41.625Z"}

```

Fig. 7. Sample JSON data of all participants fetched from the server

```

e {_label: "609d2b98ad7b6100153a0b1b", _distance: 0.38954991855464877}
  distance: 0.38954991855464877
  label: "609d2b98ad7b6100153a0b1b"
  _distance: 0.38954991855464877
  _label: "609d2b98ad7b6100153a0b1b"
  [[Prototype]]: Object

```

Fig. 8. Shortest Euclidean Distance (L2) between detected feature vectors and one of the feature vectors in the face datasets

```

if (bestMatch._label !== "unknown") {
  createTrxCallback({
    variables: {
      attendanceID: props.match.params.attendanceID,
      studentID: bestMatch._label,
    },
  });
  console.log("Saving in db now");
}
});

```

Fig. 9. Code snippet of saving the attendance transaction of recognised participant's ID

```

1. start
2. retrieve dataset from database
3. database receives query
4. database processes query
5. database returns list of Feature Vectors of all enrolled students
   in the course to system
6. receive Feature Vectors strings
7. convert Feature Vectors string to Float32Array
8. initiate image acquisition
9. initiate face detection
10. IF Bounding Box Score is more than minConfidence Score
    mark bounding boxes (detected face)
    ELSE
    IF repeat face matching process
        go to step 8
    ELSE
        end
11. initiate facial landmark detection
12. mark detected facial landmark points with dot
13. align detected bounding box
14. initiate feature extraction
15. calculate list of L2 Distance from each Feature Vectors of each
    participant
16. Find Shortest L2 Distance
17. IF Shortest L2 Distance is less than Matching Threshold
    label the face as Name and Matric No
    ELSE
    label the face as "unknown"
    IF repeat face matching process
        go to step 8
    ELSE
        end
18. record attendance (Student ID, Timestamp)
19. database receives query
20. database processes query
21. database returns list of Feature Vectors of all enrolled students
    in the course
22. receive result (success or fail)
23. IF repeat face matching process
    go to step 8
    ELSE
        end

```

**Fig. 10.** Pseudocode of Face Matching to record attendance

### 3.5 System Architecture

The section discussed the system architecture of the project, as shown in [Fig. 11](#). The project is a client-server architecture that involves the network connection to provide the application services. There are two main modules: the face datasets registration module and face matching module, which require the transfer of the weight files from the deep learning pre-trained models to perform face recognition tasks. There are three submodules in the application, which are user management, course management and attendance management. Each module is associated with API on the server, then the server interacts with the database for the particular CRUD operation.

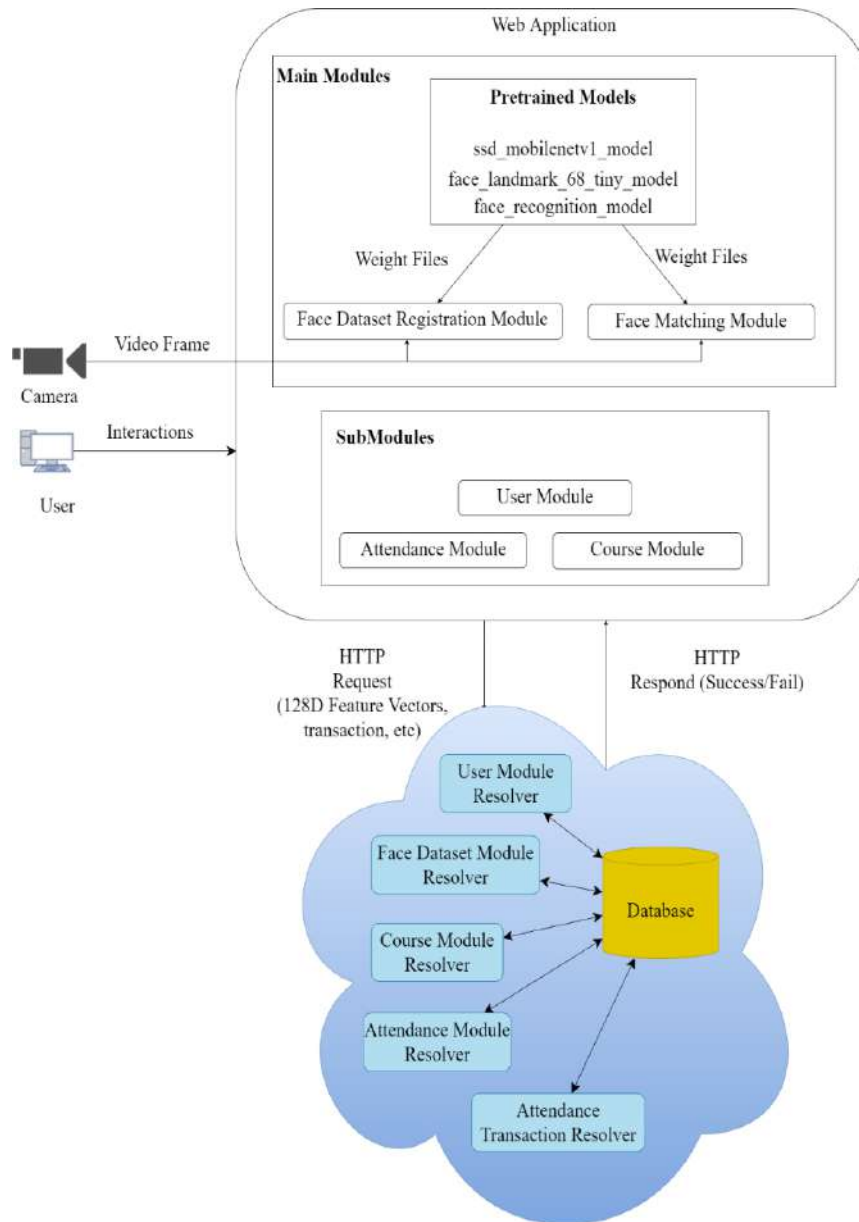


Fig. 11. System Architecture.

### 3.6 Database Design

ERD is illustrated in Fig. 12 to outline the collections and the relationship. There is a total of 6 collections to be created in MongoDB Cloud. A short description of each collection is presented in Table 4. A pre-computed feature vector for each face was used in the matching process. Therefore, only a URL reference was used to point to the actual face location in a different database. This implementation will reduce the complexity of the database and make inference faster.

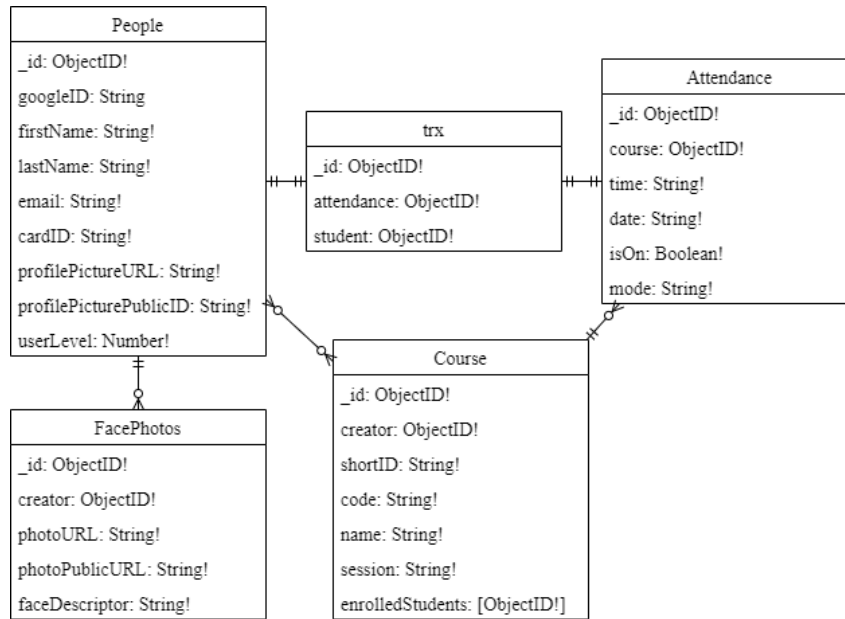


Fig. 12. ERD.

Table 4. Description of Chosen Deep Learning Pre-trained Models.

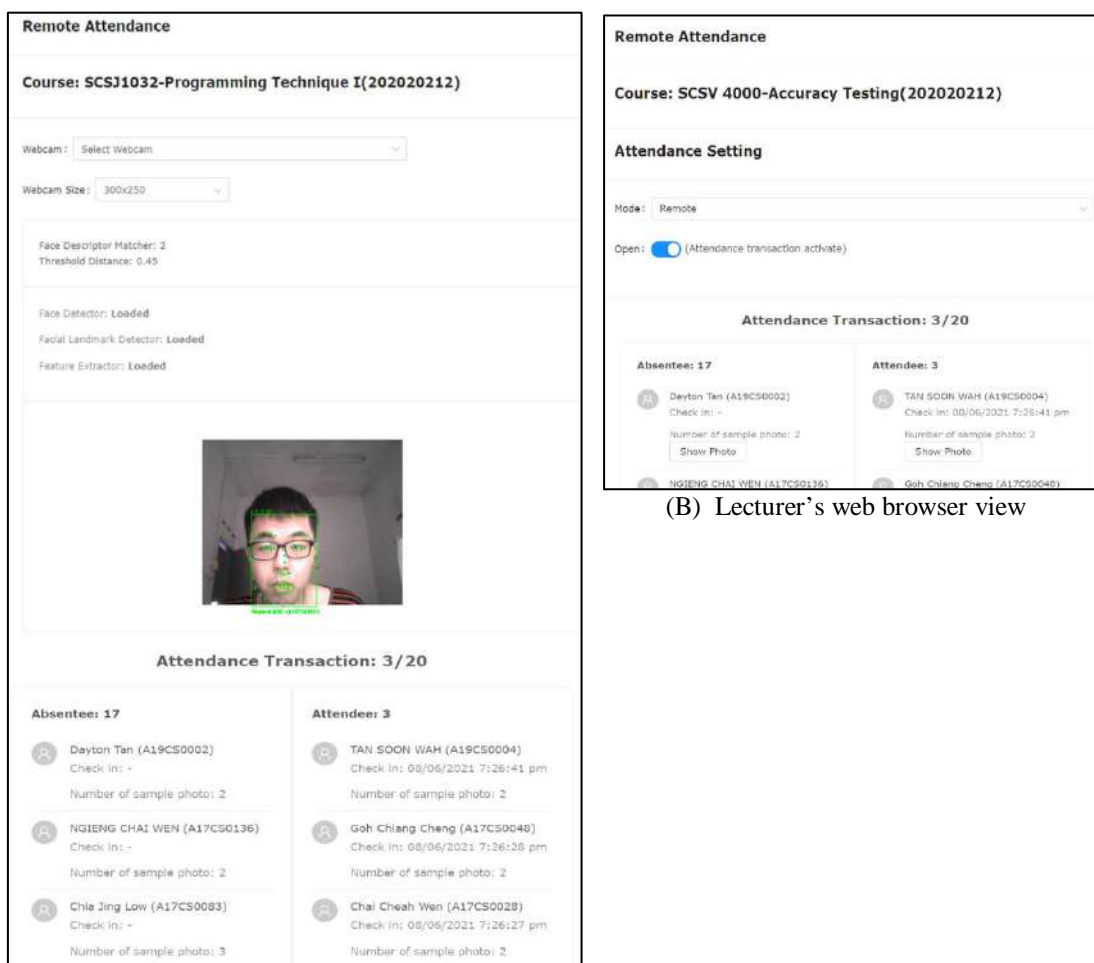
Collection	Purpose
People	Storing the user information such as name, email, password, cardID (staffID/matric number), profile picture and user level.
Face Photo	Storing the datasets, including owner ID, URL and feature vectors
Course	Storing course information such as creator ID, unique ID, course code, course name, course session and enrolled students.
Attendance	Storing attendance information such as date and time with the setting such as isOn and mode.
Trx	Storing the attendance transactions record, which is student ID and attendance ID

### 3.7 System Functionality Description

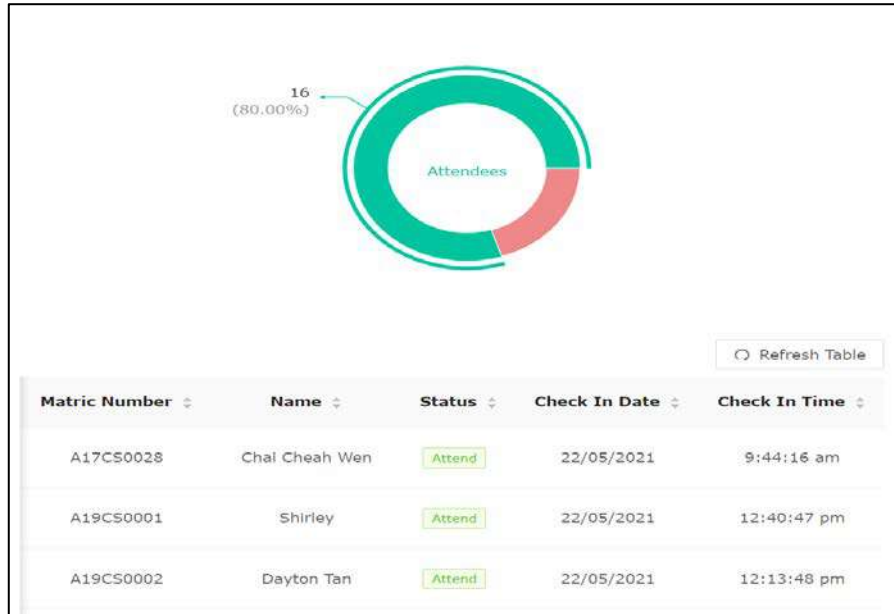
Below listed some important functions in the proposed attendance system:

- Add Course by Lecturer:** The lecturer creates a course by entering code, name, and session. A unique ten-digit course ID will be automatically assigned. Students need the course ID to enrol in the course.
- Enrol Course by Student:** Students need to enrol in the course by entering the course ID.
- Upload Face Photograph by Student:** Each user with the student role must upload at least two face photographs for the matching process. Features vectors will be generated for each uploaded photo. The system will first perform face detection to check there is only a single face; otherwise, the "Save" button will be disabled.
- Create Attendance by Lecturer:** The lecturer creates the attendance by clicking the "Create Attendance" button, then redirecting to the attendance form page. After specifying the date and time, the attendance is created after submission.

- e) Attendance Room: After the lecturer submits the attendance form, the enrolled students have to go into the attendance list and enter the attendance room. An attendance setting is provided to the lecturer who is “mode” and “open,” The webcam is opened on the student’s web browser for Remote Attendance, whereas the webcam is opened on the lecturer’s web browser for F2F Attendance (Fig. 13). When the open is on, attendance transaction is enabled and vice versa.
- f) Attendance Record: The attendance transaction details such as student name, matric number, status, check-in date, and check-in time are displayed. A visualisation chart is drawn to describe the overall class attendance (Fig. 14).



**Fig. 13.** The interface of Remote Attendance from the student (A) and lecturer view (B).



**Fig. 14.** The interface of Attendance Record.

## 4. Result and Discussions

### 4.1 Dataset Collection

Twenty respondents were invited for the performance testing on the proposed system. Before the performance testing, each respondent must register an account as the student role and upload their face images with the frontal view orientation at a minimum of two samples. A smaller number of datasets was used because the chosen deep learning pre-trained models have been trained with thousands of data; thus, trained parameters from the models are transferred and used with only a small number to achieve the same result. [Appendix 1](#) shows an example of uploaded face datasets from five of the respondents. The face datasets need to be in frontal view with unlimited face expressions. The dataset can be in various formats and types as long as it meets the frontal view requirements.

### 4.2 Performance Testing

Three tests were conducted based on pose variation such as frontal, right, left. The tests measured the outputs for five respondents in terms of the three-pose variation (e.g., frontal, right, left). The outputs for five of the respondents are demonstrated in [Fig. 15](#). The recognised face is labelled with the name, matrix number, and calculated distance (L2). The resulting L2 value is the shortest among all the spaces between detected feature vectors and the feature vectors in the datasets.



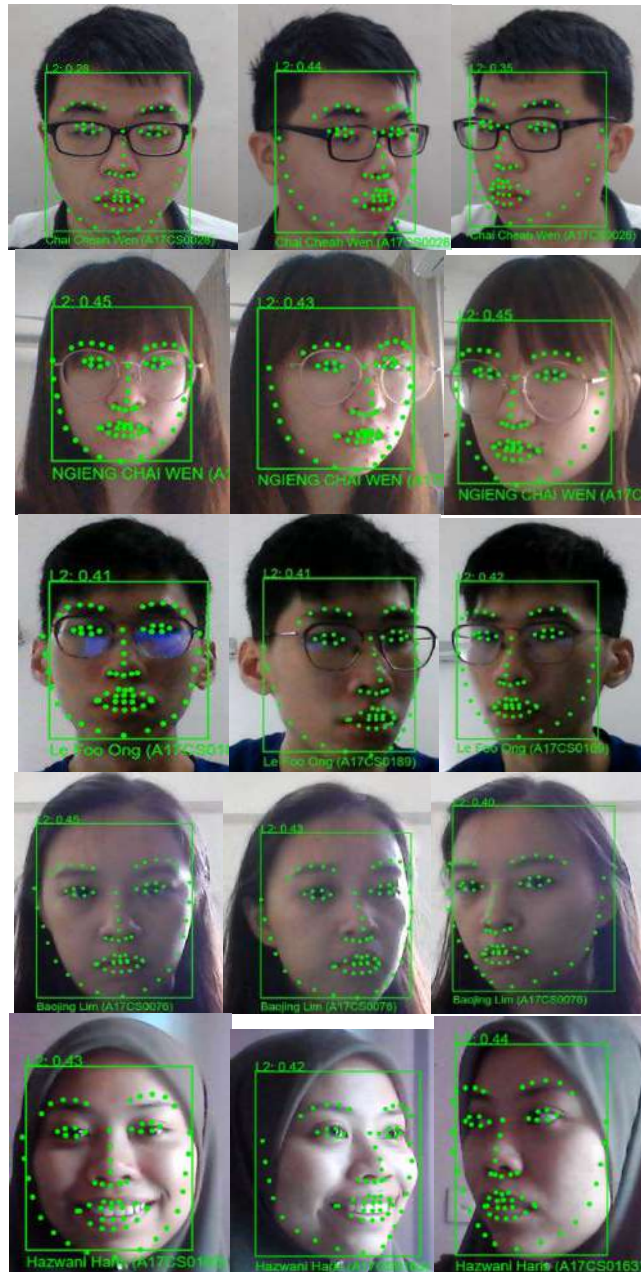


Fig. 15. Testing results (e.g., Frontal, Right, Left)

#### 4.3 Evaluation Result

Table 5, Table 6 and Table 7 illustrate the prediction values using the matching threshold of 0.45 and recorded True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) for frontal, right and left facing faces of 20 respondents. Five new subjects were chosen to test against the performance, whereby their faces were not part of the database. Hence, the testing datasets consist of a total of 25 faces. The testing is done with three times the attendance process of each subject. The testing has been done on the position of the frontal face, right face ( $\pm 30^\circ$ ) and left face ( $\pm 30^\circ$ ) at a distance between 50 to 100

cm.

The results summarised that the proposed attendance system with deep learning models successfully achieved the highest true prediction values. True Positive from 18 respondents with frontal faces, right and left facing faces were correctly predicted, meanwhile only two respondents with False Negative. **Table 8** demonstrates the overall performance based on Confusion Matrix. The proposed system achieved an overall of 92% accuracy (Eq. 1), 100% precision (Eq. 3) and 90% recall (Eq. 2).

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \times 100\% \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (2)$$

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (3)$$

**Table 5.** Testing Result (Frontal Face)

		True Value	
		True	False
Prediction Value (Threshold: 0.45)	True	TP (18)	FP (0)
	False	FN (2)	TN (5)
Accuracy	$\frac{18 + 5}{18 + 5 + 0 + 2} \times 100\% = 92\%$		
Precision	$\frac{18}{18 + 0} \times 100\% = 100\%$		
Recall	$\frac{18}{18 + 2} \times 100\% = 90\%$		

**Table 6.** Testing Result (Facing Right)

		True Value	
		True	False
Prediction Value (Threshold: 0.45)	True	TP (18)	FP (0)
	False	FN (2)	TN (5)
Accuracy	$\frac{18 + 5}{18 + 5 + 0 + 2} \times 100\% = 92\%$		
Precision	$\frac{18}{18 + 0} \times 100\% = 100\%$		
Recall	$\frac{18}{18 + 2} \times 100\% = 90\%$		

**Table 7.** Testing Result (Facing Left)

		True Value	
		True	False
Prediction Value (Threshold: 0.45)	True	TP (18)	FP (0)
	False	FN (2)	TN (5)
Accuracy	$\frac{18 + 5}{18 + 5 + 0 + 2} \times 100\% = 92\%$		
Precision	$\frac{18}{18 + 0} \times 100\% = 100\%$		
Recall	$\frac{18}{18 + 2} \times 100\% = 90\%$		

**Table 8.** Overall Result

		True Value	
		True	False
<b>Prediction Value (Threshold: 0.45)</b>	<b>True</b>	TP (18+18+18 = 54)	FP (0+0+0 = 0)
	<b>False</b>	FN (2+2+2 = 6)	TN (5+5+5 = 15)
<b>Accuracy</b>	$\frac{54 + 15}{54 + 15 + 0 + 6} \times 100\% = 92\%$		
<b>Precision</b>	$\frac{54}{54 + 0} \times 100\% = 100\%$		
<b>Recall</b>	$\frac{54}{54 + 6} \times 100\% = 90\%$		










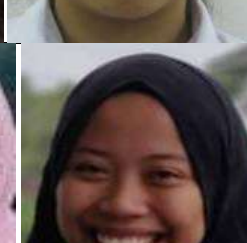
As shown in **Table 8**, the precision rate is 100% because of the strict matching value used. Thus, it can reduce the possibility of False Positive cases with little False Negative cases. However, the testing result can be varied if tested using different thresholds and distances from the camera. Since the testing is carried out individually at a distance between 50 cm to 100 cm, thus the resolution of the face region is quite high, and the captured feature vectors contain more information and are accurate. On the other hand, the testing is done remotely using a different camera and computer hardware. Hence these factors can cause slight issues on the performance testing. For future works, performance testing needs to be done using different matching thresholds to determine the desired threshold. More variation of the face, such as other resolution and head tilt, can be collected from the respondents to get a more diverse reference to boost the accuracy in a different environment. The web application can be visited at: <https://attendlytical.netlify.app> for a live demo.

## 5. Conclusion

This study presented a web-based attendance system using deep learning face recognition. The proposed attendance system utilised the state-of-the-art deep learning face recognition pre-trained models, combined them into the web environment and linked them to an online database for feature storage. The design of the proposed system implemented the procedure of face recognition which are registration of face datasets and face matching, integrated into a client-server architecture web-based system. The performance testing on the face recognition was conducted and measured in the production site to simulate the real scenario. The face datasets collected from all the respondents should have the appropriate illumination and high resolution for future improvement. Besides, several test cases should be carried out to determine the suitable threshold value to get the optimal testing results since the selected matching threshold is arbitrary.

## Appendix

**Appendix 1.** Face datasets

<b>Respondent</b>	<b>Uploaded Face Image</b>	
Respondent 1		
Respondent 2		
Respondent 3		
Respondent 4		
Respondent 5		

## Acknowledgement

The authors wish to thank the Universiti Teknologi Malaysia (UTM), VicubeLab research group and Ministry of Higher Education Malaysia for funding this study.

## References

- [1] Zhu, Y., & Jiang, Y., "Optimisation of face recognition algorithm based on deep learning multi-feature fusion driven by big data," *Image and Vision Computing*, vol. 104, pp. 1-8, 2020. [Article \(CrossRef Link\)](#)
- [2] Wati, V., Kusriani, K., Al Fatta, H., & Kapoor, N., "Security of facial biometric authentication for attendance system," *Multimedia Tools and Applications*, vol. 80, pp. 23625–23646, 2021. [Article \(CrossRef Link\)](#)
- [3] Kak, S. F., Mustafa, F. M., & Valente, P., "A review of person recognition based on face model," *Eurasian Journal of Science & Engineering*, vol. 4, no. 1, pp. 157-168, 2018.
- [4] Sinha, P., Balas, B., Ostrovsky, Y., & Russell, R., "Face recognition by humans: Nineteen results all computer vision researchers should know about," in *Proc. of the IEEE*, vol. 94, no. 11, pp. 1948-1962, 2006. [Article \(CrossRef Link\)](#)
- [5] Wang, C., Wang, Y., Chen, Y., Liu, H., & Liu, J., "User authentication on mobile devices: Approaches, threats and trends," *Computer Networks*, vol. 170, 2020. [Article \(CrossRef Link\)](#)
- [6] Yang, J., Zhang, D., Frangi, A. F., & Yang, J. Y., "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131-137, 2004. [Article \(CrossRef Link\)](#)
- [7] Gardezi, S. J. S., Faye, I., Adjed, F., Kamel, N., & Hussain, M., "Mammogram classification using chi-square distribution on local binary pattern features," *Journal of Medical Imaging and Health Informatics*, vol. 7, no. 1, pp. 30-34, 2017. [Article \(CrossRef Link\)](#)
- [8] Wiskott, L., Krüger, N., Kuiger, N., & Von Der Malsburg, C., "Face recognition by elastic bunch graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 775-779, 1997. [Article \(CrossRef Link\)](#)
- [9] Ahmed, S., Frikha, M., Hussein, T. D. H., & Rahebi, J., "Optimum feature selection with particle swarm optimization to face recognition system using gabor wavelet transform and deep learning," *BioMed Research International*, 2021. [Article \(CrossRef Link\)](#)
- [10] Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L., "Deepface: closing the gap to human-level performance in face verification," in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 1701-1708, 2014. [Article \(CrossRef Link\)](#)
- [11] Schroff, F., Kalenichenko, D., & Philbin, J., "Facenet: A unified embedding for face recognition and clustering," in *Proc. of The IEEE Conference On Computer Vision and Pattern Recognition*, pp. 815-823, 2015. [Article \(CrossRef Link\)](#)
- [12] Ahmed, S. B., Ali, S. F., Ahmad, J., Adnan, M., & Fraz, M. M., "On the frontiers of pose invariant face recognition: a review," *Artificial Intelligence Review*, vol. 53, pp. 2571–2634, 2020. [Article \(CrossRef Link\)](#)
- [13] Apoorv, R., & Mathur, P., "Smart attendance management using bluetooth low energy and android," in *Proc. of 2016 IEEE Region 10 Conference (TENCON)*, pp. 1048-1052, 2016. [Article \(CrossRef Link\)](#)
- [14] Amri, U. F., Hashim, N. N. W. N., & Hanif, N. H. H. M., "Speech-based class attendance," in *Proc. of IOP Conference Series: Materials Science and Engineering*, vol. 260, no. 1, p. 012008, 2017. [Article \(CrossRef Link\)](#)
- [15] Charity, A., Okokpujie, K., & Etinosa, N. O., "A bimodal biometric student attendance system," in *Proc. of 2017 IEEE 3rd International Conference on Electro-Technology for National Development (NIGERCON)*, pp. 464-471, 2017. [Article \(CrossRef Link\)](#)
- [16] Ayop, Z., Lin, C. Y., Anawar, S., Hamid, E., & Azhar, M. S., "Location-aware event attendance system using QR code and GPS technology," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 9, pp. 466-473, 2018. [Article \(CrossRef Link\)](#)



- [17] Shah, S. N., & Abuzneid, A., "IoT based smart attendance system (SAS) using RFID," in *Proc. of 2019 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, pp. 1-6, 2019. [Article \(CrossRef Link\)](#)
- [18] justadudewhohacks/face-api.js [Online]. Available: <https://github.com/justadudewhohacks/face-api.js/>, Accessed on June 22, 2021
- [19] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C., "SSD: Single shot multibox detector," in *Proc. of European Conference on Computer Vision*, pp. 21-37, 2016. [Article \(CrossRef Link\)](#)
- [20] He, K., Zhang, X., Ren, S., & Sun, J., "Deep residual learning for image recognition," in *Proc. of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778, 2016. [Article \(CrossRef Link\)](#)



**Nor Azman Ismail** is Associate Professor of Human-Computer Interaction (HCI) and Associate Chair Research and Academic Staff at School of Computing, Universiti Teknologi Malaysia (UTM). For the past 25 years, Nor Azman has been an active member of the computing research community. His research interests include Multimodal Interaction, Computer Vision, Social Media Analytic and Web Mining. He obtained his Bachelor of Science in Computer Science & Edu (Mathematics) from UTM, Master of Information Technology from the National University of Malaysia, and PhD in Human-Computer Interaction from Loughborough University, United Kingdom.



**Chai Cheah Wen** has been an undergraduate at the School of Computing, Universiti Teknologi Malaysia (UTM), studying in Bachelor's Degree of Computer Science (Graphics & Multimedia Software). He is interested in the field of Computer Graphics, Computer Vision and Web Technology.



**Hussein Samma** and received his Ph.D. degree in computer vision and machine learning from Universiti Sains Malaysia (2016). He has M.Eng. degree in Computer Engineering from Jordan University of Science and Technology (2009), and B.Eng. degree in computer engineering from Yarmouk University, Jordan (2006). His research interest in computer vision, deep learning, swarm intelligence, and soft bio-metrics. He is a member of IEEE computational intelligence society and he has served as a reviewer for several well-known international journals such as machine vision and applications. He is currently senior lecturer at the school of computing, faculty of engineering, Universiti Teknologi Malaysia (UTM).



**Md. Sah Bin Haji Salam** is an IEEE and VICUBELAB member at the University Technology Malaysia. He holds a PhD and Master degree from the University of Technology Malaysia in Computer Science from the School of Computing of the UTM Faculty of Engineering and a Bachelor of Science degree from the University of Pittsburgh, PA, USA. He was a visiting research in Universidad Polytechnica de Madrid, Spain under the Speech Technology Group. His research areas are speech and image processing with artificial intelligent application.





**Layla Hasan** Holds a BSc degree in computer science from the University of Jordan, Jordan; an MBA degree in business administration from the University of Jordan, Jordan; and a PhD degree in information science/ Human Computer Interaction (HCI) from Loughborough University, UK. She is currently a Senior Lecturer in the School of Computing, Faculty of Engineering at Universiti Teknologi Malaysia (UTM), and a member of Information Engineering and Behavioral Informatics (INFOBEE) research group. Her latest position was an Associate Professor in Software Engineering/Human Computer Interaction (HCI) at Zarqa University, Jordan. Besides her work as a teacher and a researcher, she worked as an editorial secretary for The International Arab Journal of Information Technology (IAJIT) for three years. She also worked as a director of the General Secretariat of the Colleges of the Computing and Information Society in the Association of Arab Universities for three years. Her research interests include human computer interaction, usability methods, web analytics, e-learning and website quality.



**Nur Haliza Abdul Wahab** is Senior Lecturer at School of Computing, Universiti Teknologi Malaysia (UTM). Nur Haliza actively involve in many research fields and her interests include Internet of Things, Augmented Reality, Indoor Positioning and Localization, Artificial Intelligent and Blockchain. She holds a PhD, Master and Bachelor in Electrical, Electronic Engineering from Universiti Teknologi Malaysia.



**Farhan** is the Deputy Director (Corporate Affairs, Publication and Research) at MaGICX, UTM. Currently, he is also an IEEE senior member, a committee member for IEEE Computer Society Malaysia, and a member for ACM MyHCI-UX Malaysia. Farhan was a member of the 100 ACM Global Practitioner Advisory Committee 2017 -2018. He was a fellow for CEO@faculty 2.0 (2017-2018) attached to Huawei Malaysia. He received the Royal Academy of Engineering United Kingdom, Leaders in Innovation Fellowship in 2017. Farhan was previously appointed as Chief Technology Officer (CTO) of MaGICX Sdn. Bhd. (2015-2017).



**Nur Eiliyah @ Wong Yee Leng** obtained her Master degree and Doctor of Philosophy in Computer Science from Universiti Teknologi Malaysia (UTM). Previously, she was a research assistant at the Department of Software Engineering, UTM, concerned with the Research and Development in the areas of Speech Recognition and Multimodal Biometric System. Currently, she is a lecturer at Applied Computing, School of Computing, Faculty Engineering at Universiti Teknologi Malaysia (UTM) and a member of Soft Computing Research Group. Her research interests include Big Data Computing, Artificial Intelligence, Pattern Recognition, Multimodal Biometric and Machine Learning.



**Mohd Foad Rohani** received his B.E (Hons.) and M.Sc. degrees in Electrical and Electronic Engineering from University Malaya (UM), Kuala Lumpur and University of Wales, Cardiff UK in 1994 and 1998, respectively. In 2013, he obtained his Ph.D (Computer Science) degrees from Universiti Teknologi Malaysia (UTM), Malaysia in the area of Network Security. Currently he is a Senior Lecturer at the School of Computing, Faculty of Engineering. He is also hold a position of Associate Chair (Facilities) and research member of Information Assurance and Information Security Research Group (IASRG). His research interest includes Pattern Recognition, Network Security, and Blockchain.