THE UNIVERSITY OF DANANG
UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF ADVANCED SCIENCE AND TECHNOLOGY

GRADUATION THESIS

DESIGN AN INTEGRATED FACIAL RECOGNITION SYSTEM FOR STAFF WELCOMING AND ADMINISTRATION IN UNIVERSITY

NGUYEN LUONG QUANG, 15PFIEV2, 122150058
LA TAN DAT, 15PFIEV2, 122150014
TRAN QUANG HUY, 15PFIEV2, 122150029
LE TIEN NHAT, 15PFIEV2, 122150051

Examiner: **Assoc. Prof. Dr. PHAM VAN TUAN**

Supervisor: **Dr. NGUYEN THI ANH THU**

Co-supervisors: **Eng. DO HOANG NHON**

Submitted to the Faculty of Advanced Science and Technology,
French-Vietnamese Excellence Engineer Training Program (PFIEV) in Industrial Informatics,
in Partial Fulfillment of the Requirements for the Degree of Engineer

Đà Nẵng, 06/2020

ABSTRACT

Facial and emotional recognition technology is not only an obvious choice for security, healthcare, business, and identification purposes, but it can also be used creatively and repurposed to serve different stakeholders. In recent years, this potential technology has been studied and applied for smart-university development. We propose an Automated Employee Attendance Management System (AEAMS) for welcoming and administrating faculty members who work at a university's building.

In this study, many facial detection techniques are considered and the Multi-Task Cascaded Convolutional Network (MTCNN) of Facenet-Pytorch pre-trained model has been selected for face detection.

Facial recognition: the feature vectors are created by using ResNet34 network. Face recognition is carried out by Support Vector Machine. Besides the welcoming front-end module, a web-based application for querying information to manage staff is also built as a back-end module. The proposed face recognition algorithm has been trained and tested on a self-built database consisting of staff who work at a building of the university. The evaluation results show high recognition rates in terms of Precision, Recall, Accuracy, F1-score, and reasonable real-time response processing speed.

Emotional recognition: the feature vectors are created by using Xception [30] network. Emotion recognition was pre-trained on the training set of the DUT_Emotion dataset by Keras framework, which was then used to predict regression and output emotional queries. Then we tested on the testing set of DUT_Emotion dataset. The evaluation results show that the recognition rate is high with the model that predicts two output states (happy and neutral).

The proposed system has been installed at the University for further development and research on face recognition and emotional recognition technology.

ACKNOWLEDGMENT

This diploma thesis was written in June, 2020 at DUT under supervision of Prof. Pham Van Tuan and Prof. Nguyen Thi Anh Thu, The Faculty of Advanced Science and Technology, PFIEV, The University of Danang, University of Science and Technology.

First of all, the authors would like to send a special thanks to Prof. Pham Van Tuan and Prof. Nguyen Thi Anh Thu for their patience, eager help and supervision in every aspect of the thesis. We also take this opportunity to express a deep regard to Mr. Le Huu Duy for the invaluable support and guidance.

We would also like to thank Mr. Do Hoang Nhon (Fablab MakerSpace Danang) for instrumental support in designing the system. Sincere thanks to the teachers in the Vietnam - France High Quality Engineer Training Program for helping us facilitate this internship. In addition, we give our gratitude to all of the staff at Smart Building for their enthusiasm in supporting us during the internship period.

We would like to give a grateful thanks to the OK Team from ECE class, FAST, DUT, we would never forget your kindness during this project. Finally, thanks to the guys in class 15PFIEV2 for making the whole thing more memorable in five years at DUT.

Danang, June 2020,

DUT: Danang University of Science and Technology

FAST:The Faculty of Advanced Science and Technology

ECE: Electronic and Communication Engineering

PFIEV: Programme de Formation d'Ingénieurs d'Excellence au Vietnam

Group authors

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATION

AEAMS	Automated Employee Attendance Management System
SVM	Support Vector Machine
CNNs	Convolutional Neural Networks
App	Application
RE	Recall
PRE	Precision
ACC	Accuracy
F1	F1- Score
MTCNN	Multi-Task Cascaded Convolutional Neural Network
ReLU	Rectifier Linear Units
ID	Identification
HOG	Histogram of oriented gradients
KNN	K-Nearest Neighbors
MQTT	Message Queuing Telemetry Transport
ResNet	Resident Neural Networks

INTRODUCTION

1. Motivation

In recent decades, increasing technological advances are reshaping every aspect of our lives, from how we produce energy to how we find and present information. Higher education institutions cannot be left behind in the use and development of these new technologies. In fact, university facilities are gradually seeking out and applying state-of-the-art technology equipment in their daily operations. Examples are intelligent management systems that apply artificial intelligence techniques. The salient fact is that face recognition and emotional recognition has been studied and implemented in many areas such as human resource management, security, health care, education and in various aspects of society. Face recognition techniques have been developed richly and quickly, allowing for more accurate recognition and real-time speed. In Vietnam, research and application of practical applications related to human face recognition are beginning to be interested in a number of areas. However, research, design and application of these types of systems to improve management effectiveness in university education institutions have not been implemented.

The research and implementation of the human face recognition and emotional recognition system to automatically monitor people inside the administrative building of The University of Danang, University of Science and Technology will help improve the efficiency of facilities in manage and operate staff, improve research and workplace security, create university information databases and serve many other practical applications. In addition to deploying scientific projects in the field of artificial intelligence, IoT applied to the unit itself will help promote research and teaching activities within the unit, consistent with the university development goals - research orientation based on industry 4.0, contributing to the development of the University into a Smart University.

2. Contribution of the thesis

The facial detectors were tested according to various evaluation criteria, from which the research team will have reference materials to apply which method to use for the recognition system according to needs and intended use.

Successful applied research: An integrated human face recognition software system that communicates with cameras and information displays installed in the administration building, University of Technology - Danang University of Technology. The software includes a welcome interface module and a web-based information management interface module. The management system with identification technology achieves a defined accuracy of > 90%.

The system has been deployed specifically for the University of Technology - Danang University. The system is installed in the administration building to handle the management of incoming and outgoing information in an intelligent and convenient way for the University, increasing the security monitoring in the building; at the same time, the function of welcoming officials and guests inside and outside is intelligent.

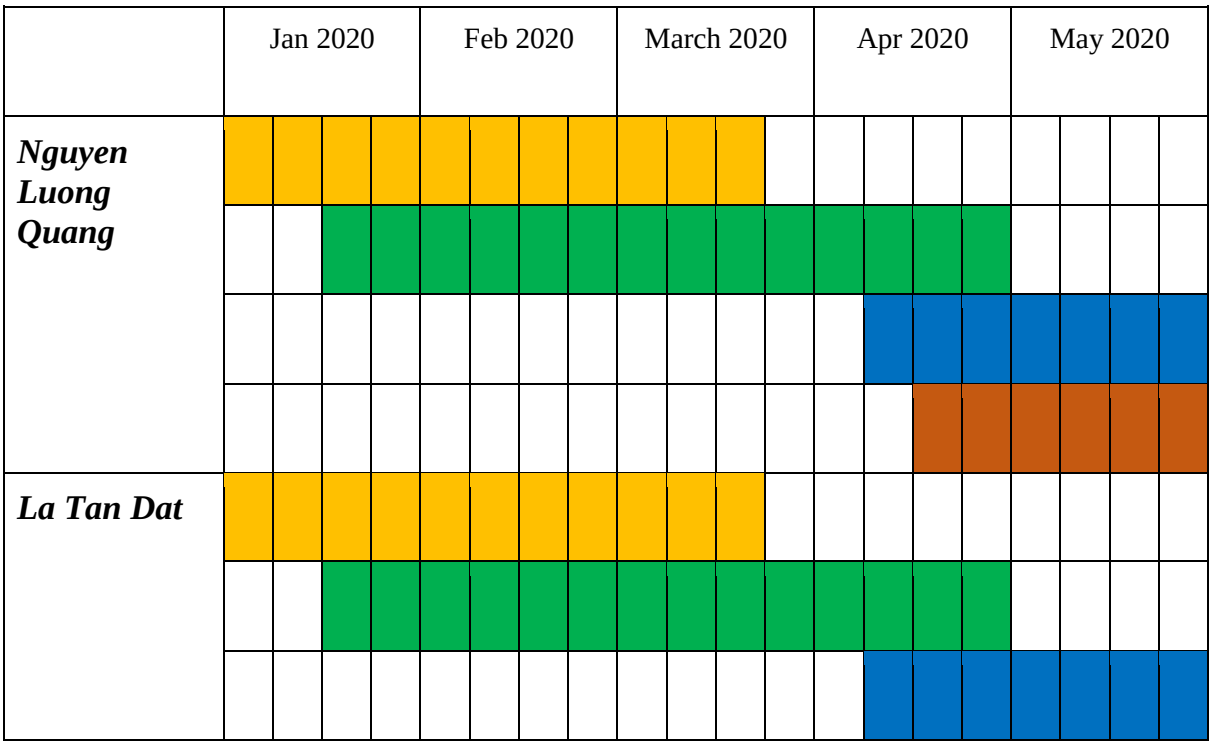
The results of the study include hardware and software, as well as a database of human faces of staff members from the functional units working in the administration building transferred to the University.

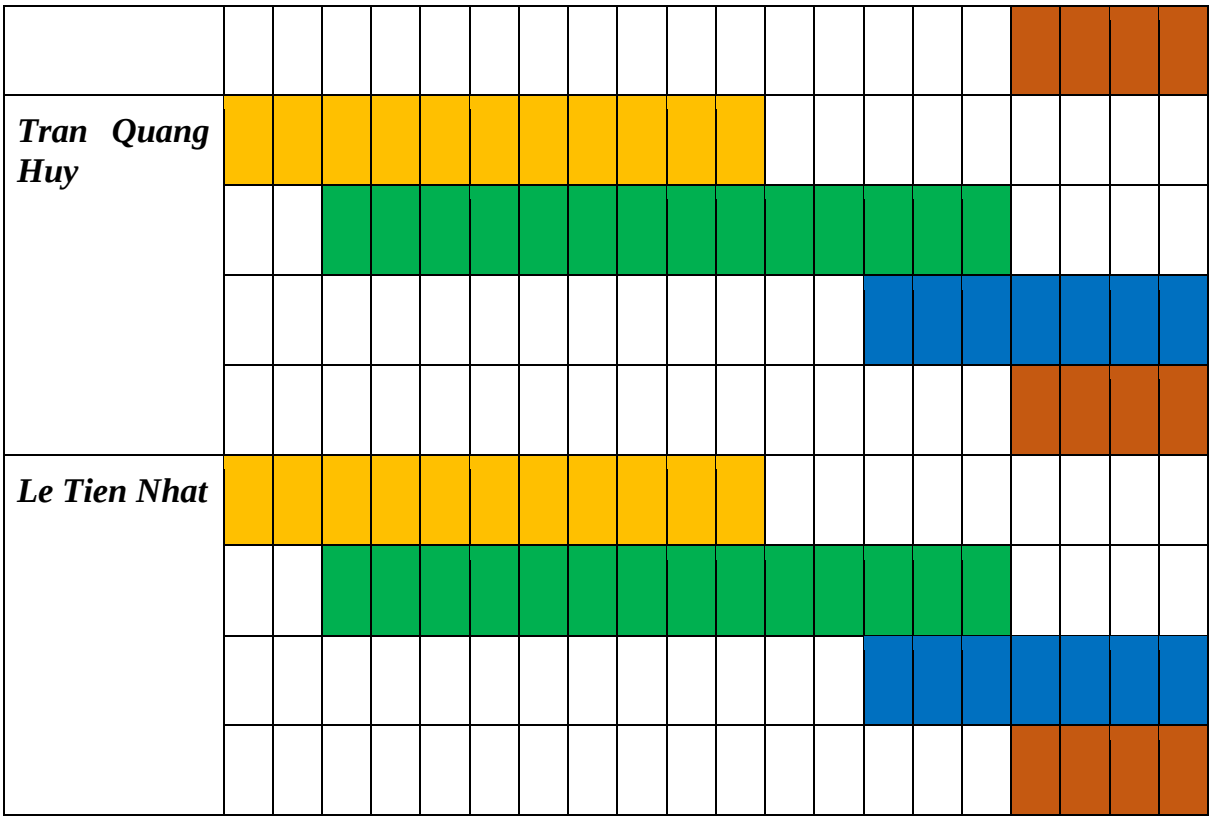
3. Organization of the thesis

The thesis "Design an Integrated Facial Recognition System for Staff Welcoming and Administration in University" which consists of 7 chapters covering the study of system installation to the application of knowledge gained in practical situations.

- ➢ *Chapter 1: Overview*
- ➢ *Chapter 2*: *Methodology*
- ➢ *Chapter 3: Databases*
- ➢ *Chapter 4: System Setup and Implementation*
- ➢ *Chapter 5: Evaluation and Experimental Results*

4. Milestone of the project





Theory Study

Design & Simulation

Evaluation

Report

5. Work distribution

1. Nguyen Luong Quang:
- Test and adjust the position of the camera for the best result
- Deploy and install the system
- Study machine learning.
- Study popular face detection methods.
- Evaluate perform of face detection module
- Study Python.

2. La Tan Dat:
- Test and adjust the position of the camera for the best result.
- Deploy and install the system
- Study Machine Learning.
- Study Python.
- Collecting and analyzing face data at Smart Building

3. Tran Quang Huy:
- Deploy and install the system
- Study Python.
- Study Machine Learning.
- Design and build the module for the system.
- Evaluate perform of face recognition module

4. Le Tien Nhat:
- Deploy and install the system.
- Study Machine Learning.
- Study PHP, MySQL, Python, Shell script, JavaScript, HTML, CSS.
- Web-App and Web-Server development.
- Developer and evaluate performance of face emotion module.

CHAPTER 1 - Overview

Chapter introduction: In this chapter, intelligent identification applications are examined with related techniques. Based on this, a proposed identification system builds on current emergencies.

1.1. Related works

Nowadays, facial recognition and emotional recognition technology is not only an obvious choice for security, healthcare, business and identification purposes, but it can also be used creatively and repurposed to serve different stakeholders. In recent years, this potential technology has been studied and applied for smart-university development such as attendance checking, participant attention tracking, satisfaction level and security checking based on face recognition and emotion recognition. We propose an integrated facial recognition and emotional recognition system for welcoming and administrating faculty members who work at a university's building.

Human facial recognition and emotional recognition has been identified as an important challenge for society and policymakers, governments and smart cities as shown in an AI Now Academy report in New York [1]. Big companies like NEC, Facebook, Google, Microsoft and many large companies are developing face recognition technology used in many of their products. Face detection is automatically locating faces in the frame or video and identifies a specific person. Many challenges facing a face recognition system, such as high accuracy, the short processing time for real-time application, the ability to identify in unfavorable conditions such as partially obscured, expressions, viewing angles, lighting conditions, etc. There are two main approaches to facial recognition: Feature-Based Face Recognition, and Appearance Based Face Recognition [2].

So far, many methods have been developed to help classify and identify faces. The Support Vector Machine (SVM) technology [3] combined with the main component analysis method Independent Component Analysis (ICA) [4] are typical techniques for high recognition performance. A study in [5] indicates that the combination of SVM and Gabor filters is well suited for changes in brightness conditions, posture and facial expressions. However, this method requires a large number of calculations, so the processing speed is quite slow. Currently, the two most widely used methods for detecting human faces in images are the Haar Cascade [6] and HOG characteristics [7]. Besides, according to another study in [8], the combination of Histogram of Oriented Gradients (HOG) and SVM has

been proved to be highly effective. Another report [9] showed that the application of pretreatment techniques also helped to increase identification rates. The detection of objects using cascade based on Haar Cascade features is an effective and fast method of object detection using simple features Boosted Cascade [10], [11].

Recent years have shown that CNN can potentially outperform all classic approaches based on standard features due to its generalization ability. The approach that CNN utilized [12] was quite similar to a classic one: the window with learned features was sliding over the pyramid of input images and the resulted data fed to a fully-connected layer. Another way of constructing a face detection system was proposed in [13] where authors suggested using the inherent multi-scale, pyramidal structure of DCNN to build feature pyramids. Mini – Xception [14] development from Xception model [15] which is a model is proposed by Francois Chollet. Xception is an extension of the inception Architecture which replaces the standard Inception modules with depth-wise Separable Convolutions.

Apart from the classification mentioned above, face detection networks can be categorized into two classes: single-shot and multi-shot detectors. One of the most well-known examples of the single shot detector is SSD network [16], which takes an image as an input and computes a feature map with bounding boxes for each class. A similar approach was utilized in [17]. Multi-shot detector (usually two-stage) suggests using several stages. The stages usually include proposal steps and refinement steps. One of the most well-known examples of the two-stage detector is R-CNN [18]. The network extracts region proposals with the selected search algorithm and then warps cropped proposals to a square. Features are calculated based on a sparse set of candidates, and the output is fed to a classifier.

The networks with a pure cascade architecture held leading positions for a long time in WIDER FACE challenge [19]. One of the most widely-used was the MTCNN detector [20], which performs both face detection and face landmarking. The network uses three sub-networks: P-Net, R-Net and O-Net. The first stage does the coarse face detection producing proposal regions. Then the Non-Maximum Suppression algorithm reduces the number of overlapping boxes forming more certain regions, which are fed to R-Net. R-Net refines the selected proposals, and O-Net does the face landmarking. MTCNN is still proposed to be used in the state-of-the-art face recognition system described in [20]. Moreover, it is utilized in the most popular public face recognition implementation of Facenet [21] available on GitHub.

Besides excellent performance, MTCNN is a promising network in terms of robustness to adversarial attacks. The first shallow P-Net has a receptive field size

of 12x12, which makes the detection resistant to the small artefacts on a face.

In recent times with the development of deep machine learning techniques, the ResNet34 neural network [22] has been developed to effectively serve to extract facial features. ResNet34 relies on a face image to extract 128D vectors that are specific to all facial features. These feature vectors are then labeled and serve to match and predict the input image. One of the simplest and most effective matching techniques is to measure Euclidean distances on multi-dimensional attribute vectors [23].

From the advantages of the techniques analyzed and to address the challenges identified above, a facial recognition model combines techniques - Multi-Task Cascaded Convolutional Networks, ResNet34 and Euclid distance measurement - have been selected to study and analyze the effectiveness of identification in complex identification conditions such as different aspects of faces, light directions, number of identities, etc., and assessing the processing speed Real-time response logic.

1.2. Automated Employee Attendance Management System proposal

The proposed AEAMS consists of many modules as shown in Figure 1.1:

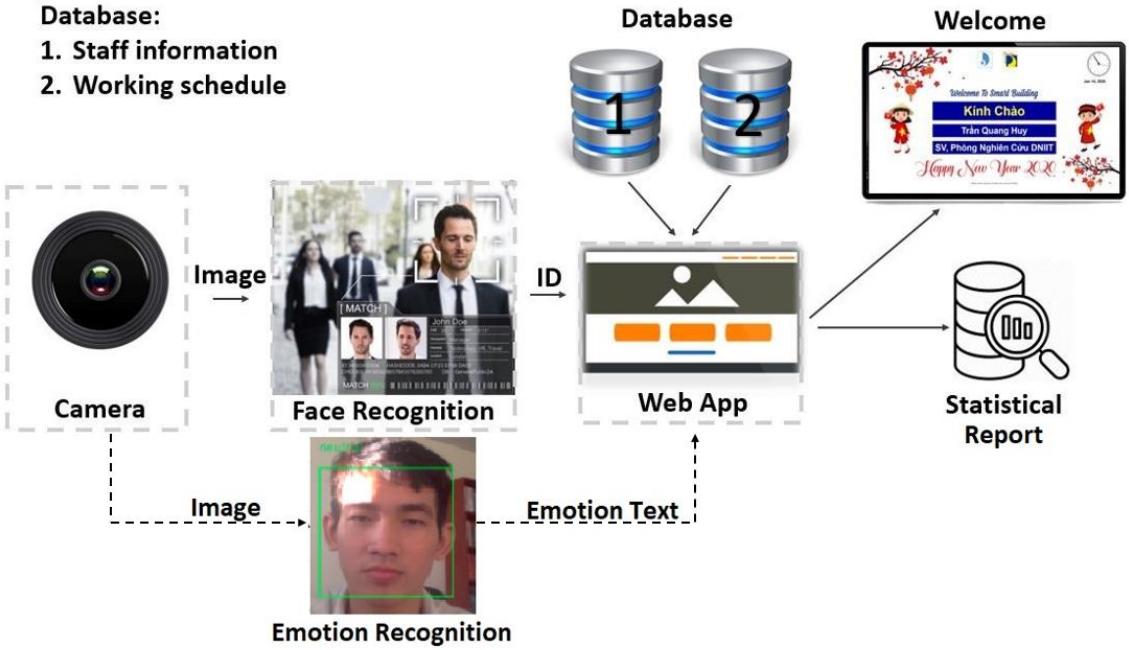


Figure 1.1 Block diagram of the integrated facial recognition system

The specific operating process of this system is: Firstly, the camera takes each frame from the video when it detects that someone has entered the door and face images will be cropped. Then they will go through the Facial recognition system

and the output will be ID, emotion. This ID will be matched to existing datasets including Staff lists and timetables. The emotional recognition module being studied with input will also take the image from the camera and the resulting text message will appear on the screen. Finally, display welcome information via the screen (Figure 1.2) and at the same time the system stores the image and the entrance time. You can use Web App (Figure 1.3) to access this dataset.

Figure 1.2 Welcoming screen

Figure 1.3 Web app interface

1.3. Face Recognition proposal

The core of the system is the face recognition module. The facial recognition module proposed in this study includes the steps shown in Figure 1.4.

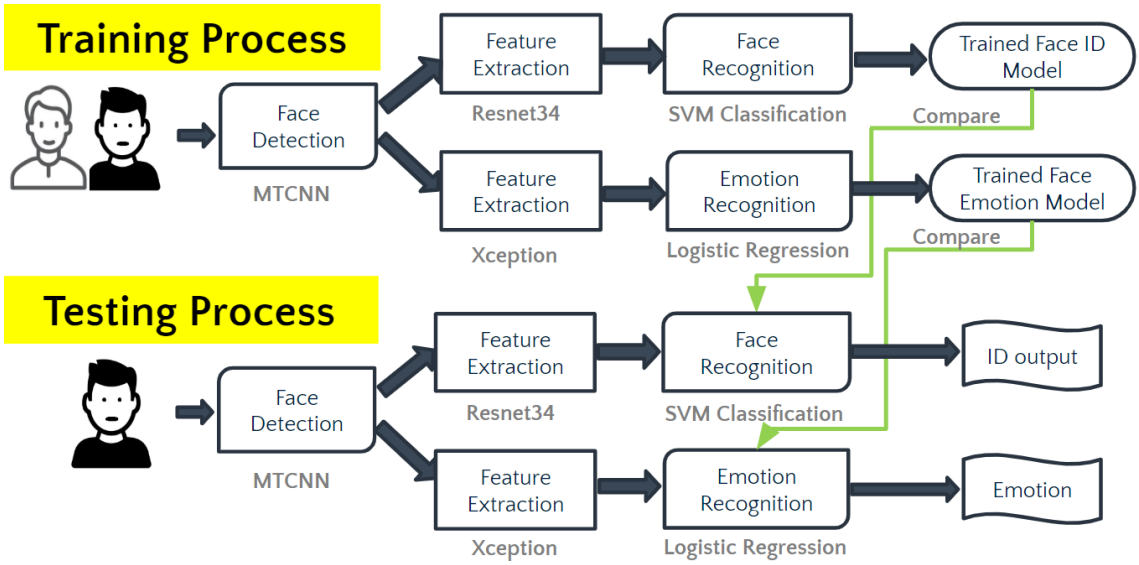


Figure 1.4 General diagram of recognition process

The first is the training phase, the input image is taken from the Self-built dataset. The face is then detected by MTCNN and the feature extraction step will be performed by ResNet34 to extract the prominent features of the face. These steps are all carried out in the early stages of training and testing. However, at the training stage, a set of vectors created specifically for each trained face. These vectors are labeled with names to facilitate identification at a later stage. During the testing phase, the decoding process will rely on the characteristic vectors that have been previously trained to make predictions.

1.4. Emotion recognition proposal

The emotional recognition module proposed in this study includes the steps shown in Figure 1.4. The first is the training phase, the emotional dataset has been trained according to the DUT_Emotion dataset (will be introduced in Chapter 3). The face is then detected by MTCNN in the training phase, however in the testing phase is detected by MTCNN and the feature extraction step will be performed by Xception [15] to extract the prominent features of the face emotions. These steps are all carried out in the early stages of training and testing. However, at the training stage, a set of vectors created specifically for each trained face. These vectors are labeled with names to emotional identification at a later stage. During the testing phase, the decoding process will rely on the characteristic vectors that have been previously trained to make predictions.

Chapter conclusion: Through the chapter, the design of our proposed AEAMS has been considered, which includes two main modules, Emotion Recognition and Web Application.

CHAPTER 2 - Methodology

Chapter introduction: In this chapter, all the methods and techniques related to the project are introduced. The chapter gives all the methods and techniques and how we apply them to solve the problem of the project.

2.1. Face detection

Face detection is the first step for our system. Firstly, the camera takes each frame from the real-time video when it detects that someone has entered the door. Faces are detected and cut out from all frames by using MTCNN of Facenet-pytorch pre-trained model as a face detector on batches of images. These face images are then pushed into the face recognition module for subsequent system processes.

The face detection module is also used to build the Self-built DUT dataset introduced in Chapter 3, section 3.2.2.2.

We will then introduce the methods that our team has studied and why we chose the method using MTCNN of Facenet-pytorch pre-trained model as a face detector on batches of images. Four face detection techniques are studied. They will then be compared on the criteria of accuracy to choose the most suitable technique. In the process of building the face detection model, the dataset UTK was used for testing the examined algorithms. Then, we test the processing speed of four detection packages to choose the best method that suits the system. Four detection packages using MTCNN will be considered on live recorded video. The face detection package with the fastest computational time will be chosen.

2.1.1. Face detection using Haar Cascade

The Haar Cascade algorithm is based on the Viola-Jones algorithm used in the face detection block [5]. Initially, the algorithm needs a lot of positive images (face images) and negative images (images without faces) to form a classifier. The extraction of features from that classifier is then performed. For this, the Haar features shown in Figure 2.1 are used, called a convolutional kernel. Each feature is a unique value obtained by subtracting the total number of pixels from the white rectangle from the total number of pixels below the black rectangle.

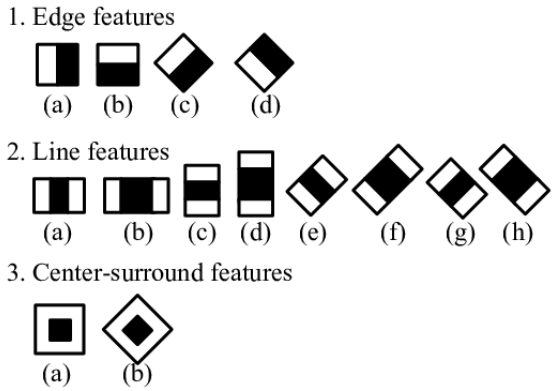


Figure 2.1 Haar Cascade feature blocks [5]

Then all the possible dimensions and positions of each kernel are used to calculate a range of features. (Even a 24x24 window results in over 160000 features). For each typical calculation, we need to find the total number of pixels under the white and black rectangle. This simplifies the calculation of the total number of pixels, how large the number of pixels may be, for an operation involving only four pixels.

2.1.2. Face detection using HOG and SVM

Generation of Histogram of oriented gradients (HOG) features using sliding windows is described in [6]. For classifying the sliding windows, the authors train a linear Support vector machine (SVM). Firstly, this method crops the faces from the entire frames. Then, applying the sobel filter to see only the edges of the faces and after that, the HOG features of the Region of Interest – face (ROI) (Figure 2.2) are extracted and the classification is done with the SVM. The model is built out of 5 HOG filters – front looking, left looking, right looking, front looking but rotated left, and a front looking but rotated right.

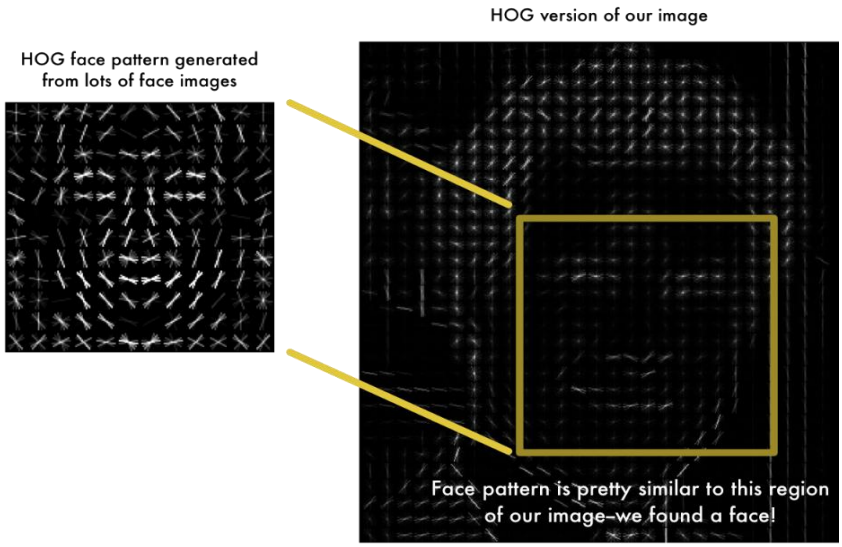


Figure 2.2 Example of HOG features of the Region of Interest [6]

2.1.3. Face detection using CNN

Convolutional Neural Networks (CNNs) which are very similar to ordinary Neural Networks, are made up of neurons that have learnable weights and biases [24]. The CNN consists of multiple layers where each layer takes a multi-dimensional array of numbers as input and produces another multi-dimensional array of numbers as output which becomes the input of the next layer [24]. When classifying images, the input to the first layer is the input image, while the output of the final layer is a set of likelihoods of the different categories [24]. The CNNs contain convolutional layer, pooling layer, fully connected layer and output layer. This is a very high level of network with different activation functions, penalties and soft-max functions involved.

A better method of CNN is to use Dlib. Dlib is a program of the OpenCV library, which supports face recognition. The algorithm that Dlib uses is HOG (Histogram of oriented gradients) and SVM (Supported vector machine). Recently, Dlib also provided additional facial recognition functions based on CNN. In this study, we use Dlib based on Faster R-CNN [25].

This method uses a Maximum-Margin Object Detector (MMOD) with CNN based features. The training process for this method is very simple and you don’t need a large amount of data to train a custom object detector.

2.1.4. Face detection using MTCNN

MTCNN is popular because it is considered as state-of-the-art face detection in [20]. This model consists of 3 separate networks: Proposal Network (P-Net), Refine Network (R-Net), and Output Network (O-Net) as depicted in Figure 2.3

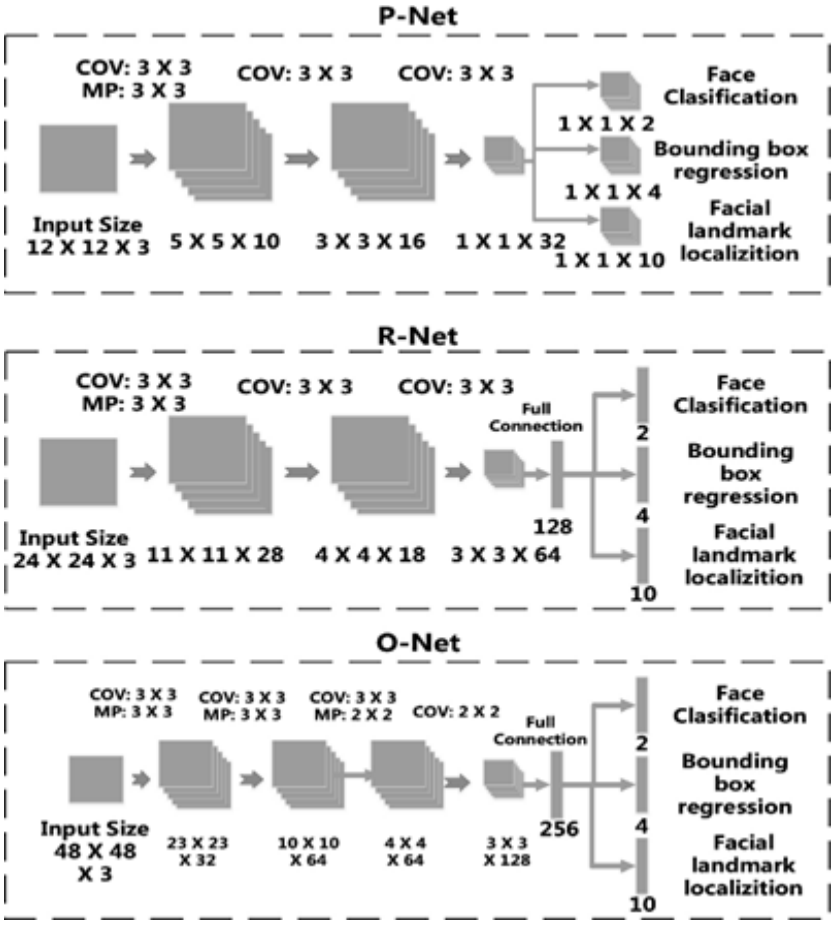


Figure 2.3 Multi-Task Cascaded Convolutional Neural Network [20]

The network creates an image pyramid for every image so that they have the image in multiple scales. In P-Net, for each scaled image, a 12x12 kernel runs through the image, searching for a face. Within each of these 12x12 kernels, 3 convolutions are run through with 3x3 kernels. After every convolution layer, a ReLu layer is implemented. After the third convolution layer, the network splits into two layers. The activations from the third layer are passed to two separate convolution layers, and a soft-max layer after one of those convolution layers. The probability of a face in each bounding box is obtained as Convolution 4–1 output, and convolution 4–2 outputs the coordinates of the bounding boxes. R-Net has a similar structure, but with even more layers. It takes the P-Net bounding boxes as its inputs and refines its coordinates. Similarly, R-Net is divided into two layers in the end, giving two outputs: coordinates of the new bounding boxes and the reliability in each bounding box. Finally, O-Net takes the R-Net bounding boxes as inputs and marks down the coordinates of facial landmarks. O-Net splits into 3 layers in the end, giving 3 different outputs: the probability of a face in the box, the coordinates of the bounding box, and the coordinates of the facial landmarks (locations of the eyes, nose, and mouth), according to [20]. The block diagram

below taken from the paper [20] provides a helpful summary of the three stages from top-to-bottom and the output of each stage left-to-right. (Figure 2.4)

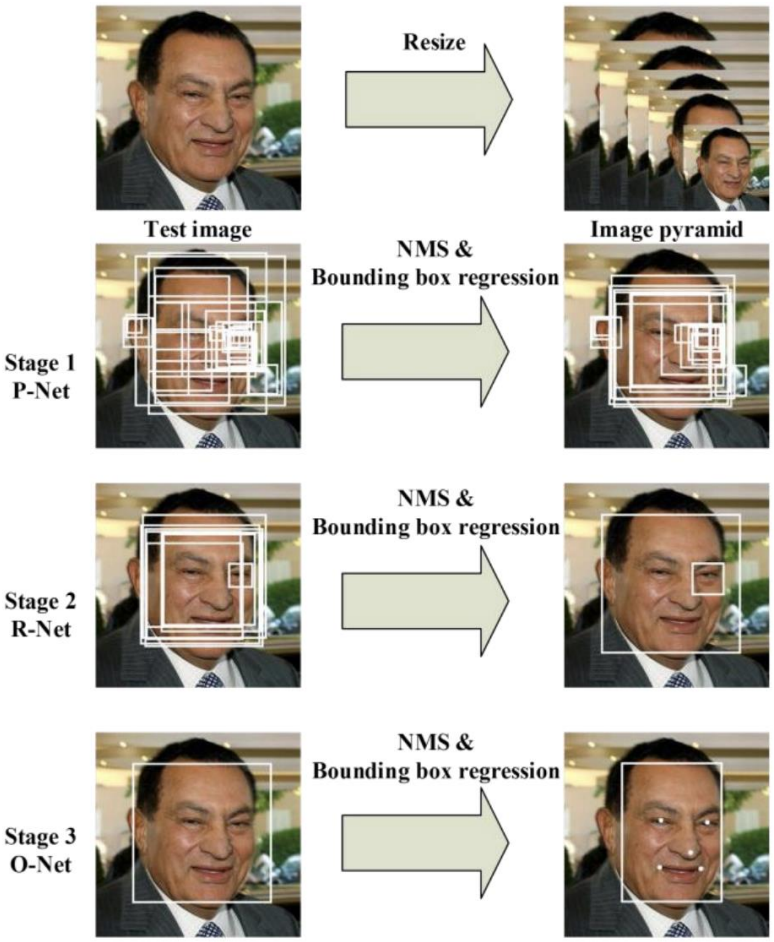


Figure 2.4 The block diagram of The MTCNN framework [20]

The model is called a multi-task network because each of the three models in the cascade (P-Net, R-Net and O-Net) are trained on three tasks, e.g. make three types of predictions; they are: face classification, bounding box regression, and facial landmark localization.

The three models are not connected directly; instead, outputs of the previous stage are fed as input to the next stage. This allows additional processing to be performed between stages; for example, non-maximum suppression (NMS) is used to filter the candidate bounding boxes proposed by the first-stage P-Net prior to providing them to the second stage R-Net model.

The MTCNN architecture is reasonably complex to implement. Thankfully, there are open source implementations of the architecture that can be trained on new datasets, as well as pre-trained models that can be used directly for face detection. For example, application of the MTCNN using Keras framework from the input image is shown in Figure 2.5.

Figure 2.5 MTCNN face detection

A better method of MTCNN is to use a Facenet-pytorch pre-trained model. Facenet is a TensorFlow implementation of the face recognizer described in [21]. Pytorch is a repository for Inception ResNet (V1) models in pytorch, pre-trained on VGGFace2 and CASIA-Webface. Pytorch model weights were initialized using parameters ported from David Sandberg's tensorflow facenet repo. Also included in this repo is an efficient pytorch implementation of MTCNN for face detection prior to inference. These models are also pre-trained. To our knowledge, this is the fastest MTCNN implementation available.

By default, the MTCNN module of Facenet-pytorch applies fixed image standardization to faces before returning so they are well suited for the package's face recognition model. Facenet-pytorch is also capable of performing face detection on batches of images, typically providing considerable speed-up. A batch should be structured as a list of PIL (Python Imaging Library) images of equal dimension. The returned object will have an additional first dimension corresponding to the batch. Each image in the batch may have one or more faces detected.

2.1.5. Face detection using Faceboxes

FaceBoxes is a CPU real-time face detector with high accuracy (Figure 2.6). It is a novel face detector, with superior performance on both speed and accuracy [26]. Specifically, this model has a lightweight yet powerful network structure that consists of the Rapidly Digested Convolutional Layers (RDCL) and the Multiple Scale Convolutional Layers (MSCL). The RDCL is designed to enable FaceBoxes to achieve real-time speed on the CPU. The MSCL aims at enriching the receptive fields and discretizing anchors over different layers to handle faces of various scales. Besides, it has a new anchor densification strategy to make different types of anchors have the same density on the image, which significantly improves the recall rate of small faces. As a consequence, the proposed detector runs at 20 FPS

on a single CPU core and 125 FPS using a GPU for VGA-resolution images. Moreover, the speed of FaceBoxes is invariant to the number of faces.

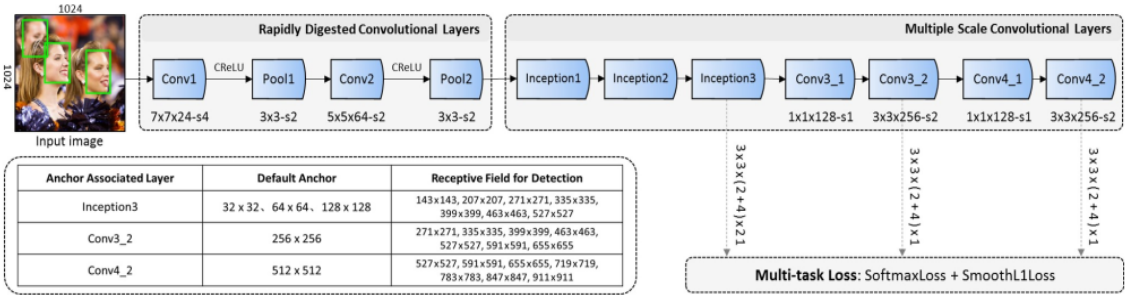


Figure 2.6 Architecture of the FaceBoxes [26]

2.2. Feature extraction and face recognition

2.2.1. Feature extraction

Nowadays, there are many available methods for facial feature extraction. Deep learning based approach is one of them which prove its performance through many research and application. The most recent state-of-the-art model architecture is Resident Neural Networks for image recognition [22].

The Resident Neural Networks, usually called ResNet, are constructed of building blocks. Each building block have the following form:

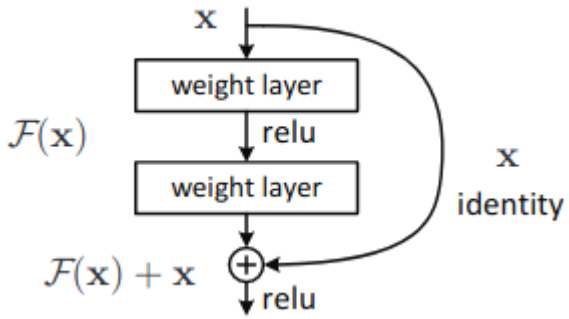


Figure 2.7 Structure of a building block [22]

Instead of hoping each few stacked layers directly fit a desired underlying mapping, we explicitly let these layers fit a residual mapping. Formally, denoting the desired underlying mapping as H(x), we let the stacked nonlinear layers fit another mapping of F(x)

$$F(x) = H(x) - x \qquad (1)$$

The original mapping is recast into F(x)+x. We hypothesize that it is easier to optimize the residual mapping than to optimize the original, unreferenced

mapping. To the extreme, if an identity mapping were optimal, it would be easier to push the residual to zero than to fit an identity mapping by a stack of nonlinear layers. The formulation of F(x)+x can be realized by feedforward neural networks with "shortcut connections" (Figure 2.7). Shortcut connections [2, 34, 49] are those skipping one or more layers. In our case, the shortcut connections simply perform identity mapping, and their outputs are added to the outputs of the stacked layers (Figure 2.7)

The layer architecture of the ResNet used in the project is shown in Figure 2.8.

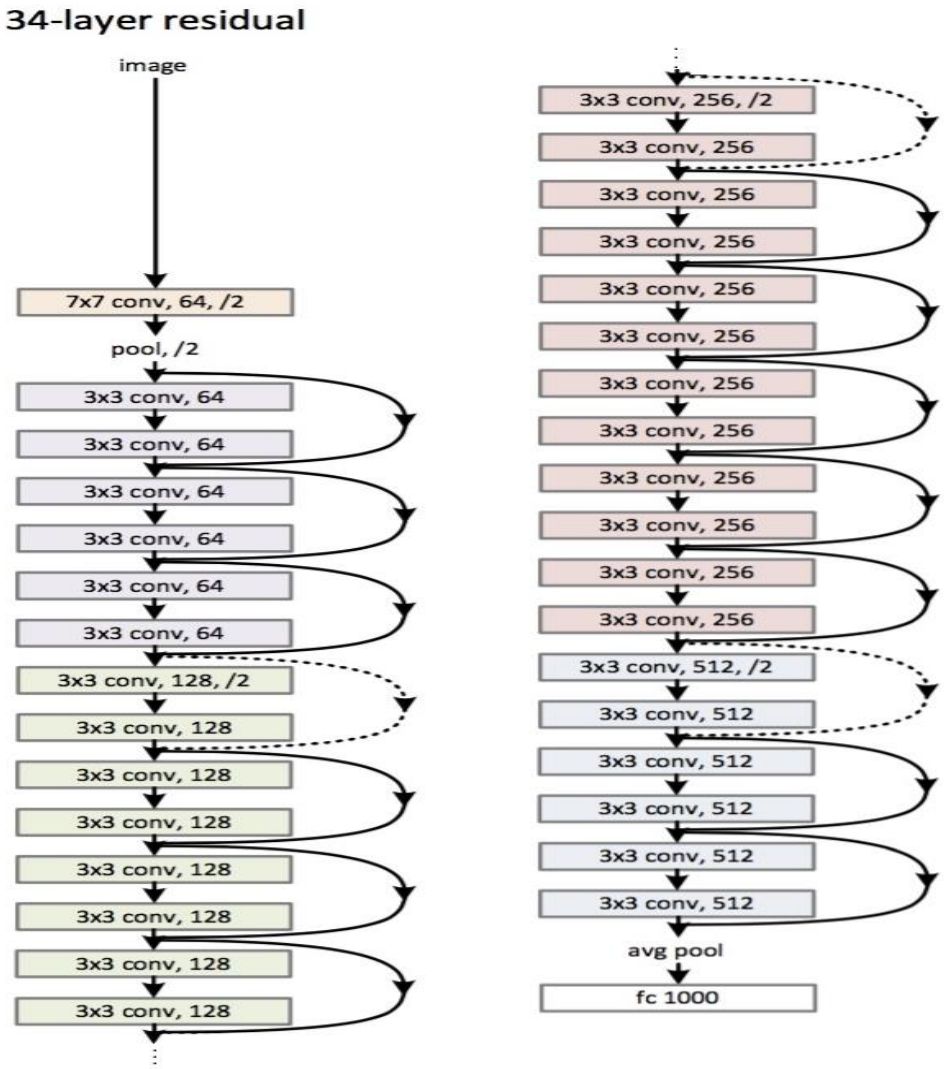


Figure 2.8 ResNet34 layer architecture. [22]

To train the networks effectively, it needs a lot of data and computing power which is limited because of the resources available for this project. Therefore, an open-source pre-trained model is our option. We use a pre-trained model on 3 million faces collected from many face datasets (LFW, FERET, etc) and the Internet. The model performance on the LFW dataset is 99.65% [22].

For each 128x128 pixels of face image input to the model, it will output a 128-dimensional feature vector. This vector is unique for each individual. For new faces that are not trained in the pre-trained model, the model will estimate a vector which is the closest to a vector of a trained person in the trained dataset. Because the trained dataset is large enough, it is always supposed to be a unique vector for each individual.

The only constraint of this model is it is trained on European and American faces datasets, although the model is applied almost on Asian face input data. This can lead to a performance reduction.

2.2.2 Recognition algorithms

After building the feature extraction block, we need an algorithm to classify this vector into which class. The below section will introduce methods to do so. All the feature vectors used in this session are computed with the feature extraction mentioned in the session 2.2.1.

The Euclidean distance and centroid method is trained and tested on FERET, Extended Yale Face B, AT&T, FASTData and self-built DUT datasets. The K-Nearest Neighbors and Support Vector Machine is trained and tested only on the self-built DUT dataset because of limited time.

2.2.2.1 Centroid estimation using Euclidean distance

To measure similarity between two feature vectors, distance measurement could be used. In Machine Learning, there are many distance measurement equations: Manhattan distance, Hamming distance, Cosine distance and Euclidean distance. Research indicates that Euclidean distance gives outstanding performance for machine learning models

$$D_e = \sqrt{\sum_{i=1}^{n}(p_i - q_i)} \qquad (2)$$

Where:

n = number of dimensions

pi, qi = data points

Multiple feature vectors of each person in the training set must have a representation which can describe all the feature vectors. In the centroid point method, this representation is a point in the center of all other points. To find this point, we can use the average calculation.

For all face images of each person in the training set, we compute a centroid point which represents the person. The trained model contains centroids points of all persons in the training set.

The following steps presents the process of training the model:

Step 1: Feature vectors and their labels are extracted from the training set.

Step 2: The first feature vector of a class will be assigned as the centroid feature vector of the class. From the second feature vector and over, the centroid feature vector is the average value of the feature vector and the current value of centroid feature vector.

Step 3: After going through all the feature vectors, there will be an average centroid feature vector for each class in the training set. Save these vectors corresponding to their labels for further us

One big advantage of this method is the very fast computational time and fast training time. In contrast, the performance of the model using this method is not as good as other methods and the model is sensitive to noise. Another constraint is to classify a person not in the training set, we must use a threshold. This threshold can only be chosen by experimenting and the threshold varies though the dataset.

2.2.2.2 K-Nearest Neighbors algorithm

The principle behind nearest neighbor methods is to find a predefined number of training samples closest in distance to the new point, and predict the label from these. The number of samples can be a user-defined constant (k-nearest neighbor learning), or vary based on the local density of points (radius-based neighbor learning). The distance can, in general, be any metric measure: standard Euclidean distance is the most common choice. Neighbors-based methods are known as *non-generalizing* machine learning methods, since they simply "remember" all of its training data. [27]

Neighbors-based recognition is a type of instance-based learning or non-generalizing learning: it does not attempt to construct a general internal model, but simply stores instances of the training data. Recognition model is computed from a simple majority vote of the nearest neighbors of each point: a query point is assigned the data class which has the most representatives within the nearest neighbors of the point. Example in Figure 2.9.

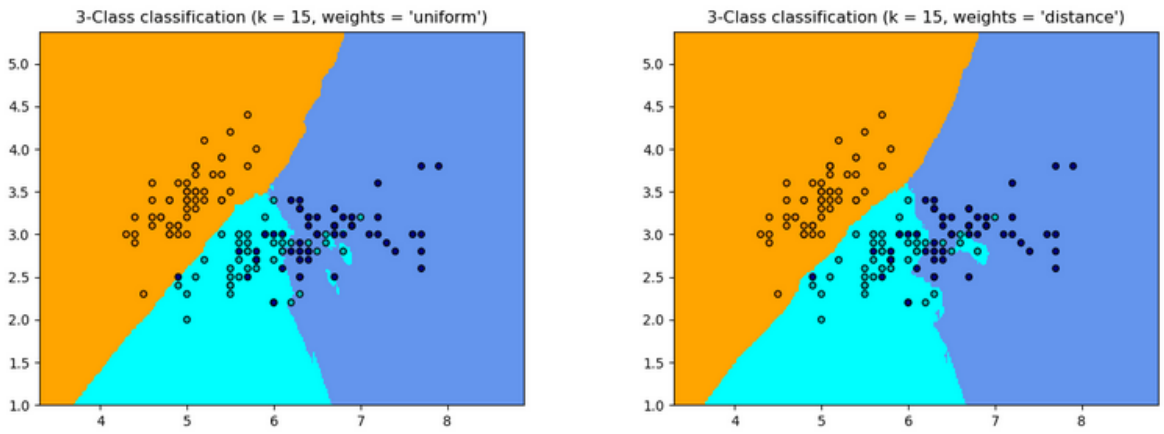


Figure 2.9 KNN classifier of 3 classes data example [27]

To train a good KNN model to work with our data, we must choose the correct number of neighbors (K). This number is data dependent. With our training set, each person contains 700 face images, we choose K = 300.

The KNN mode is trained through the steps bellow:

Step 1: Feature vectors and their labels are extracted from the training set.

Step 2: Build the KNN with the following configuration: learning algorithm is KD Tree, number of neighbors is 300, distance metric is Euclidean distance, weight metric is uniform which means different distances between neighbors have the same weight.

Step 3: Fit the feature vectors and the labels to the KNN. The algorithm will find a query point for each data class.

Step 4: After the training is done, we have a model which is the representative of the training set.

The training process of KNN models is very fast because the model only remembers the data points, not actually doing many calculations. Because of that, it takes much time for the model to predict a feature vector. About the model's performance, it gives much higher evaluation's scores than the Euclidean distance method.

2.2.2.3 Support Vector Machine algorithm

The basic idea is to find a hyperplane which separates the d-dimensional data perfectly into its two classes. However, since example data is often not linearly separable, SVM's introduce the notion of a "kernel induced feature space" which casts the data into a higher dimensional space where the data is separable. Typically, casting into such a space would cause problems computationally, and

with overfitting. The key insight used in SVM is that the higher-dimensional space doesn't need to be dealt with directly (as it turns out, only the formula for the dot-product in that space is needed), which eliminates the above concerns. [28]

Choosing the hyperplane must maximize the margin between data points and the hyperplane. Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.

Popular kernels are: Linear Kernel, Polynomial Kernel, Gaussian Kernel, Radial Basis Function (RBF), Laplace RBF Kernel, Sigmoid Kernel, Above RBF Kernel, etc.

The Regularization Parameter (usually called C) tells the SVM optimization how much you want to avoid miss classifying each training example.

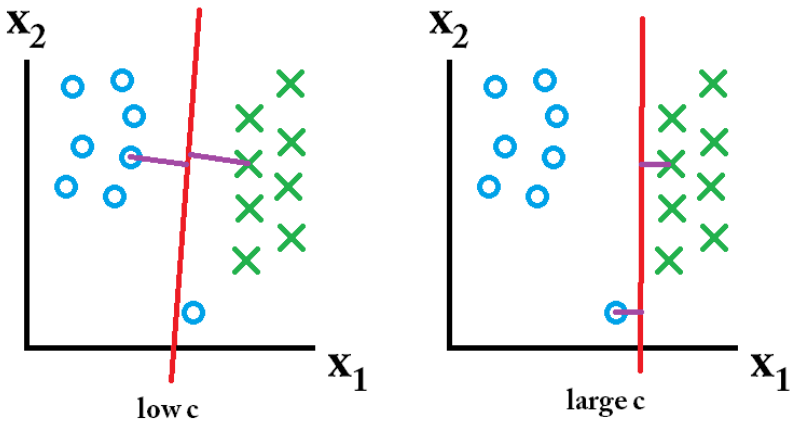


Figure 2.10 The influence of regularization parameter C

See Figure 2.10 to have an intuitive understanding. The O and X symbols are the dataset with 2 classes, the red line is the hyperplane and the purple line describes the magnitude of margin. If the C is higher, the optimization will choose a smaller margin hyperplane, so training data misclassification rate will be lower. On the other hand, if the C is low, then the margin will be big, even if there will be missed classified training data examples.

The following is the steps to train the SVM model:

Step 1: For each face image in the training set, we extract the feature vector and assign it a label corresponding to the ID of the person. After looping through the training set, we have a list of feature vectors and a corresponding list of labels.

Step 2: We build the SVM with 6 following configurations:

Configuration 1: Linear kernel, probability enabled, C=0.01, tolerance=0.001, max_iter=300.

Configuration 2: Linear kernelkernel, probability enabled, *C*=1, *tolerance*=0.001, *max_iter*=300.

Configuration 3: Linear kernel, probability enabled, C=100.1, tolerance=0.001, max_iter=300.

Configuration 4: Radial Bias FunctionLinear kernel, probability enabled, C=0.01, tolerance=0.001, max_iter=300.

Configuration 5: Radial Bias FunctionLinear kernel, probability enabled, C=1, tolerance=0.001, max_iter=300.

Configuration 6: Radial Bias FunctionLinear kernel, probability enabled, C=100, tolerance=0.001, max_iter=300.

Step 3: The list of features vectors and list of labels are fit to the SVM algorithm. It will start to find the support vectors that construct hyperplanes that separate the feature vectors the most. SVMs do not directly provide probability estimates, these are calculated using a five-fold cross-validation. If the tolerance is reached or passing through a *max_iter* number of iteration, the training is stopped and the model is generated.

Step 4: The SVM model consisting of support vectors is saved to a binary file for further usage.

SVM models perform well on high dimension feature vectors. Its speed and performance are both better than the KNN method but it needs much more time to train the model. SVM models have many parameters, to have a good model, we must correctly choose them.

The trained model evaluation results will be shown in Chapter 5.

2.2.3 Post-processing

Giving the final decision on the first sight of a face image is quite risky. We aren't sure if the detected face is actually a face or some unknown objects, if it is in good quality or blurred. To overcome this problem, the model needs to make decisions on a sequence of faces, not a single face. This will reduce the rate of poor condition samples and increase the. But in a frame, there are one or more individuals. Grouping faces belonging to the same person is essential to do the post-processing. (Figure 2.11).

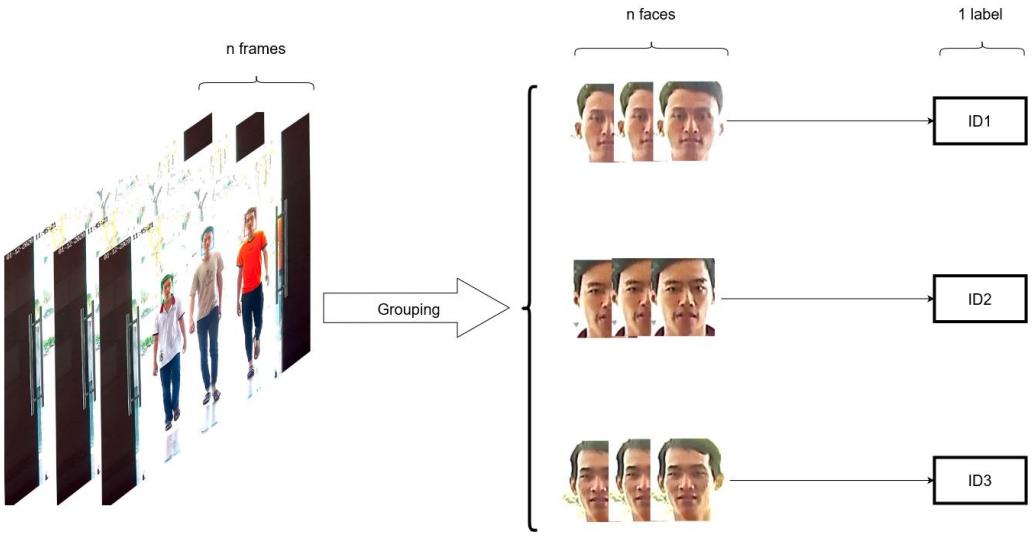


Figure 2.11 The process of post-processing.

To do this, we use the previous feature vector as a measurement of similarity for each face in the frame. The same person must have features vectors closest to each other. We use Euclidean distance proposed in 2.2.2.1 for similarity measurement.

The process of grouping faces is described as following steps:

Step 1: Read a frame in the video stream, detect faces in the frame and extract the feature vector for each face.

Step 2: Each detected face in the first frame is assigned to a group.

Step 3: For each recurrent frame, do the same process in Step 1. For each face in the frame, we measure the similarity between the feature vector of this face and the features vectors of groups. We assign the face to the group which is the most similar (minimum Euclidean distance) to the face.

Step 4: Each group has a live time and a position which is represented by the last face of that group. If the live time or the position exceed predefined values, the group is removed from the available group. In the project, we define live time as 3s, this is the average time for a person to cross the recognition zone and the position limit is 70% of the height of the frame, this is the value before the face of the incoming person cannot be detected.

After walking through these steps, we have a list of groups. For each group in the available groups, we only make decisions if the number of faces in that group is more than a number. This number is chosen by the frame rate and the performance of the face detection model. In this project, all the process from reading frames to recognized labels takes around 0.2s, this is 5 frames per second

(FPS). We give the final prediction every 1.5s according to 7 faces in a group. Because there will be some miss-recognition, so if more than a half of faces are classified as the same person, we are confident about the label and we can use the label.

2.3. Emotion recognition

2.3.1. Feature extraction of emotion

Our architecture is a fully-convolutional neural network that contains 4 residual depth-wise separable convolutions where each convolution is followed by a batch normalization operation and a ReLU activation function. The last layer applies a global average pooling and a softmax activation function to produce a prediction. This architecture has approximately 60,000 parameters; which corresponds to a reduction of 10x when compared to our initial naive implementation, and 80x when compared to the original CNN. Figure 2.12 displays our complete final architecture which we refer to as mini-Xception.

Residual modules modify the desired mapping between two subsequent layers, so that the learned features become the difference of the original feature map and desired features. Consequently, the desired features H(x) are modified in order to solve an easier learning problem F(X) such that:

$$H(x) = F(x) + x \qquad (3)$$

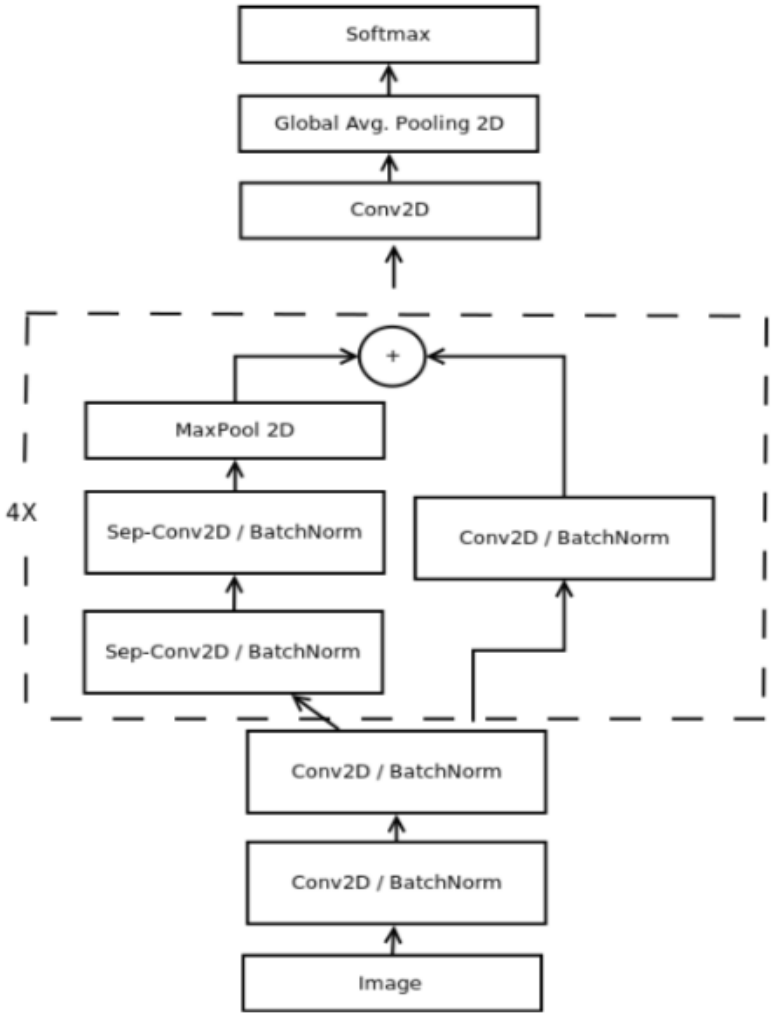


Figure. 2.12 Mini-Xception model for real-time classification [15]

Mini-Xception model has pre-trained on the DUT_Emotion dataset. Classify facial expressions from 240 examples of 48x48 pixel images of faces. Images are categorized based on the emotion shown in the facial expressions (happy, neutral). (Figure 2.13)

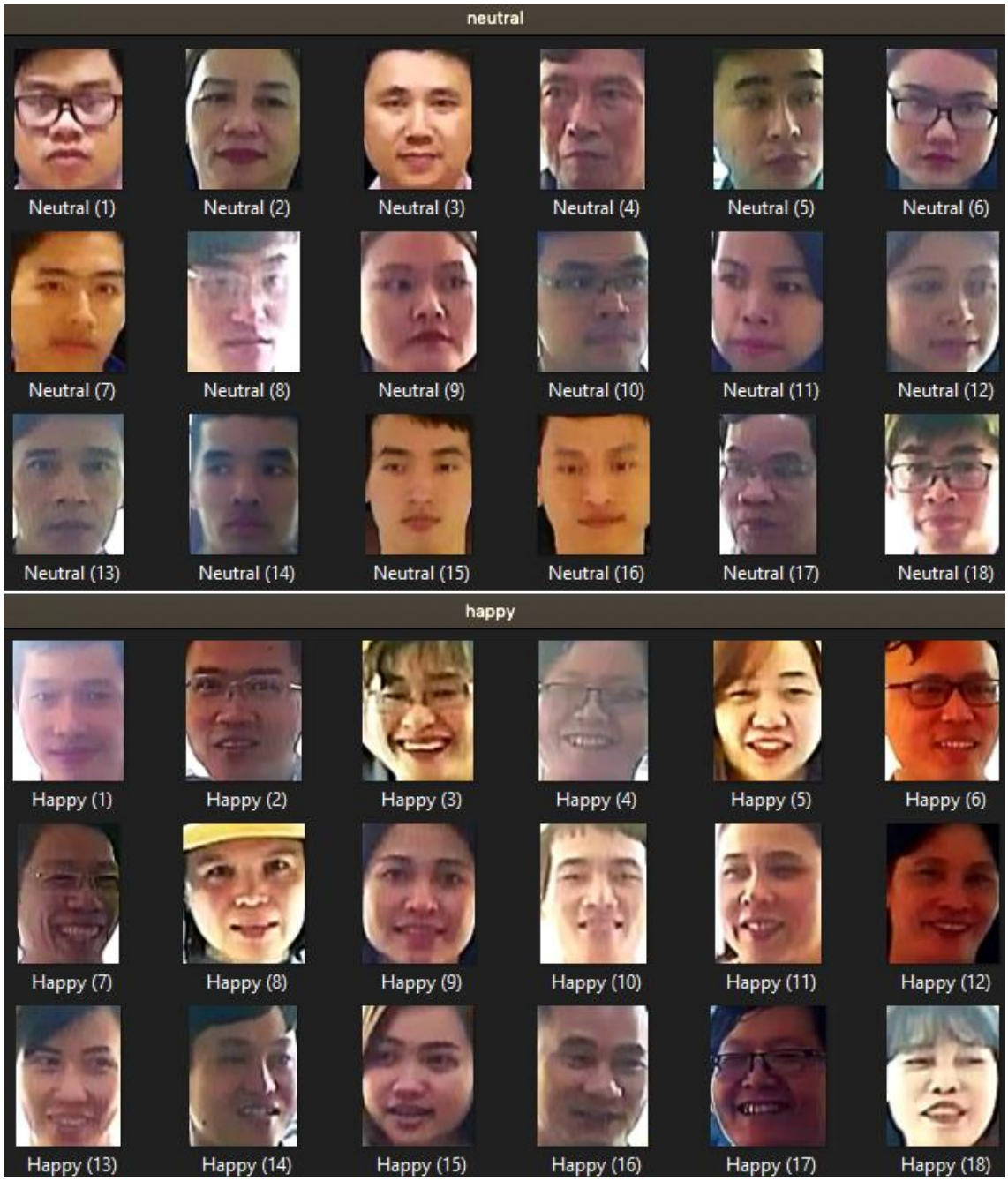


Figure 2.13. A part of DUT_Emotion dataset

2.3.2 Emotion recognition

This module uses Logistic Regression [29] to emotion recognition. Details is shown in the Figure 2.14, first processing the face vector to gray and resizing so that the output goes into the model to perform regression prediction process of (1, 64, 64, 1) after the prediction process, will produce the value of a 7-dimensional array with predicted value (gradient) in the range 0 to 1 and then take the index at biggest value of that array, continue to output the value of the value labeled DUT_Emotion dataset.

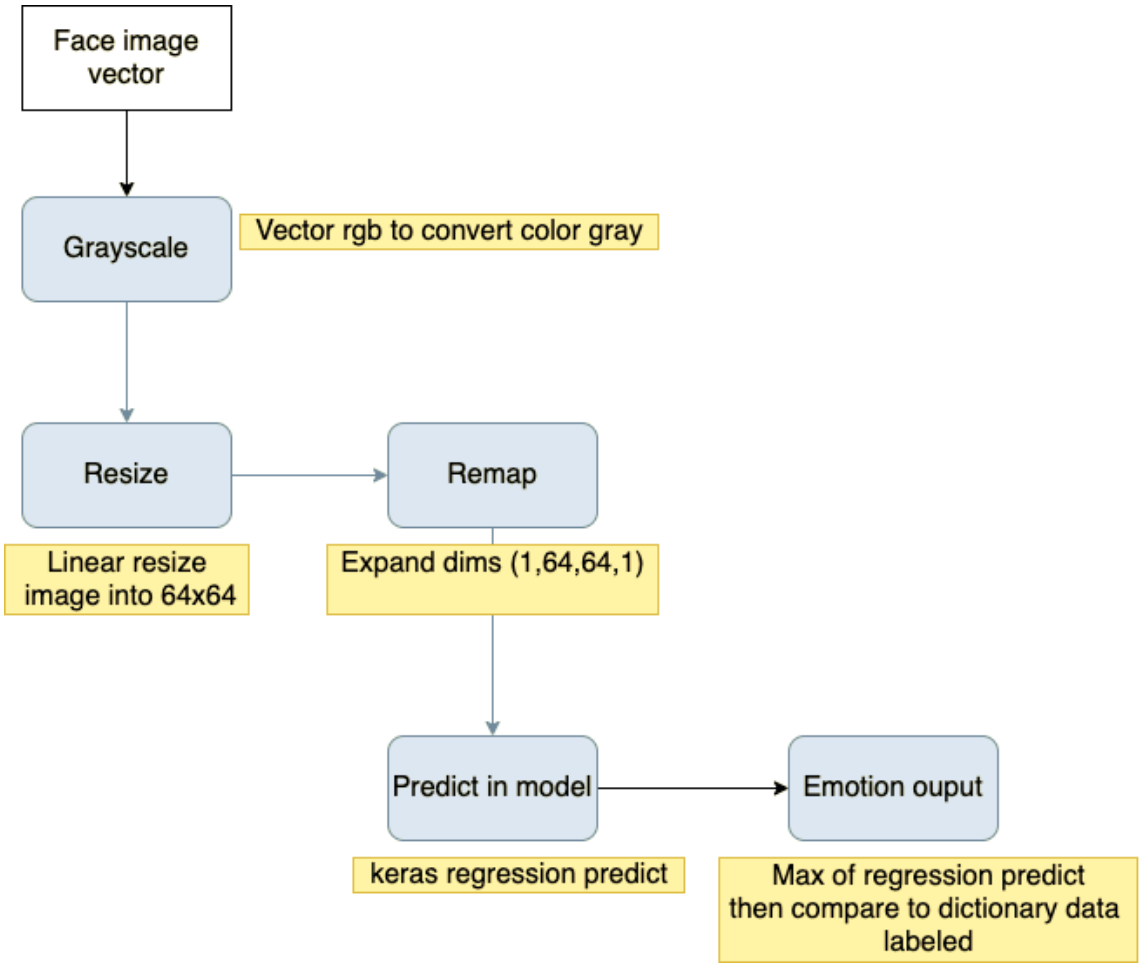


Figure. 2.14. Structure diagram emotion recognition.

Chapter conclusion: Through the chapter, all the proposed methods for the system have been considered including face recognition, feature extraction, and emotion recognition.

CHAPTER 3 - Databases

Chapter introduction: In this chapter, the evaluation is implemented on 4 available datasets (UTK, AT&T, FERET and Extended Yale B) and 2 self-built dataset: FASTData dataset and DUT dataset.

3.1 Facial online dataset

3.1.1 The UTK face dataset [30]

This dataset consists of 20000 face images in the wild which covers large variation in pose, facial expression, illumination, covered face (occlusion), resolution, etc. There is only a single face in one image, so if the result has at least 2 faces from a single image, then we know that the detector is making a mistake. That is the reason why we choose this dataset to compare the 4-face detection algorithms. The UTK face dataset is used to assess detection performance of four studied techniques mentioned in section 2.1.

Figure 3.1 The UTK face dataset [30]

3.1.2 The FERET Dataset [31]

This dataset includes 14,126 images of 1199 individuals collected from 1993 to 1996 and contains 365 sets of duplicate images. A duplicate set is the second set of images a person has in the dataset and is usually taken on another day. For some individuals, they are taken again every year and with each subject being photographed multiple times. This one helps train and test in large numbers and change the appearance of individuals. The whole dataset is then divided into two sub-datasets: Training set and Testing set. Each person has about 10 images, 7 images are used for the training set and 3 images are used for the testing set. This

test is named Test 1: A large number of individuals.

Figure 3.2 Face images from the FERET Dataset [31]

3.1.3 The Extended Yale Face Dataset B [32]

This dataset consists of 28 individuals, each with 594 gray images. The dataset was taken from various lighting angles. Each image of an object has different expressions or lighting angles: central light, left light, right light, no glasses, glasses, normal, fun, sad, sleepy, surprised and winked. The whole dataset is then divided into two sub-datasets: Training set and Testing set. The training set contains 416 images and the testing set includes 178 images from 28 persons. This dataset is used to test performance on different lighting conditions and different expressions of each individual. Thus, this test is named Test 2: Different lighting situations.

Figure 3.3 Face images from the Extended Yale Face Dataset B [32]

3.1.4 The AT&T Dataset [33]

This dataset is collected from 40 individuals, 10 images per person and contains many face poses (Center, Half Left, Half Right, Head Up, Head Down) with different emotions on the face. The whole dataset is then divided into two

sub-datasets: Training set and Testing set. Each person has about 10 images, 7 images are used for the training set and 3 images are used for the testing set. The AT&T database and FASTData, a self-built dataset, are used to check the performance of face recognition with different positions of the face. This test is named Test 3: Different positions of the face.

Figure 3.4 The AT&T Dataset [33]

3.2 Self-built dataset

3.2.1 The Self-built FASTData dataset

This dataset is a self-built dataset consisting of 50 PFIEV students, 10 images per student with different positions of faces in natural light. The FASTData dataset was also converted to grayscale format to ensure objectivity with AT&T dataset. This dataset is also divided into two sub-set: for each individual, there will be 7 images for the training set and 3 for the testing set. This is used to test in Test 3 with different facial poses, similar to the AT&T database.

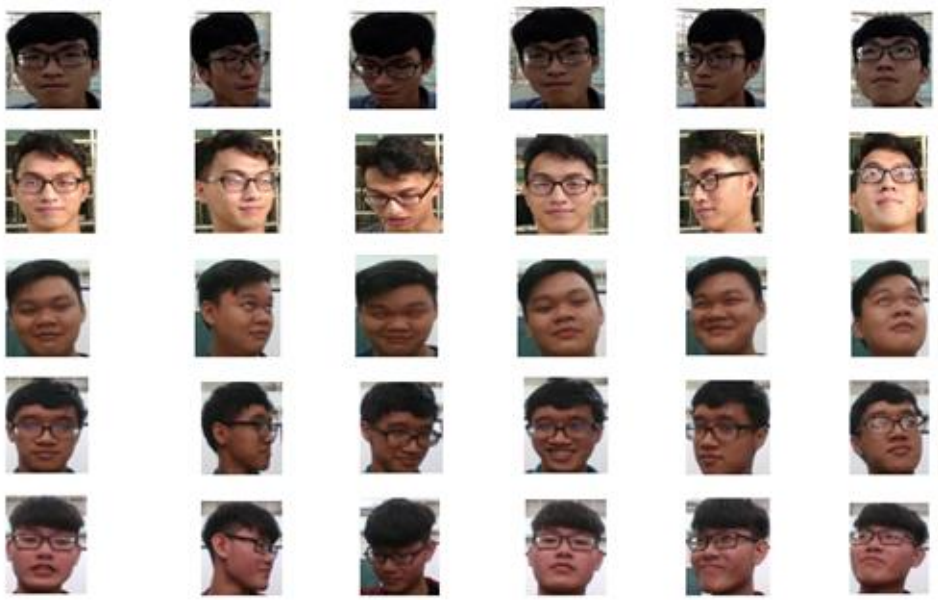


Figure 3.5 The FASTData dataset.

3.2.2 The Self-built DUT dataset

This dataset was built by capturing in real-time condition within 5 months in December 2019 and from April to July 2020. This dataset is a set of face images of 80 officials and lecturers working at the Smart Building while entering the main door with different weather conditions.

3.2.2.1 Collect video data

The process begins as soon as we have tested the locations and found the right place for the structure of the building. Camera position is located opposite the main entrance. There are many different weather conditions such as bright sunshine, rainy or normal weather. The recording time of the day is morning and afternoon (working time). We look forward to each person's set covering all his/her various possible face expressions, postures, and light conditions that might be possible at recognition time. This data is collected naturally with situations that come from all directions of moving objects. (Figure 3.6)

The stage before going to the door:
- go from left path to the entrance as A, B
- go straight to the entrance as C
- go from right path to the entrance as D, E

After entering the door:
- go to the left like direction 1 (face tilted to the left)
- go straight ahead like direction 2
- go to the right like direction 3(face tilted to the right)

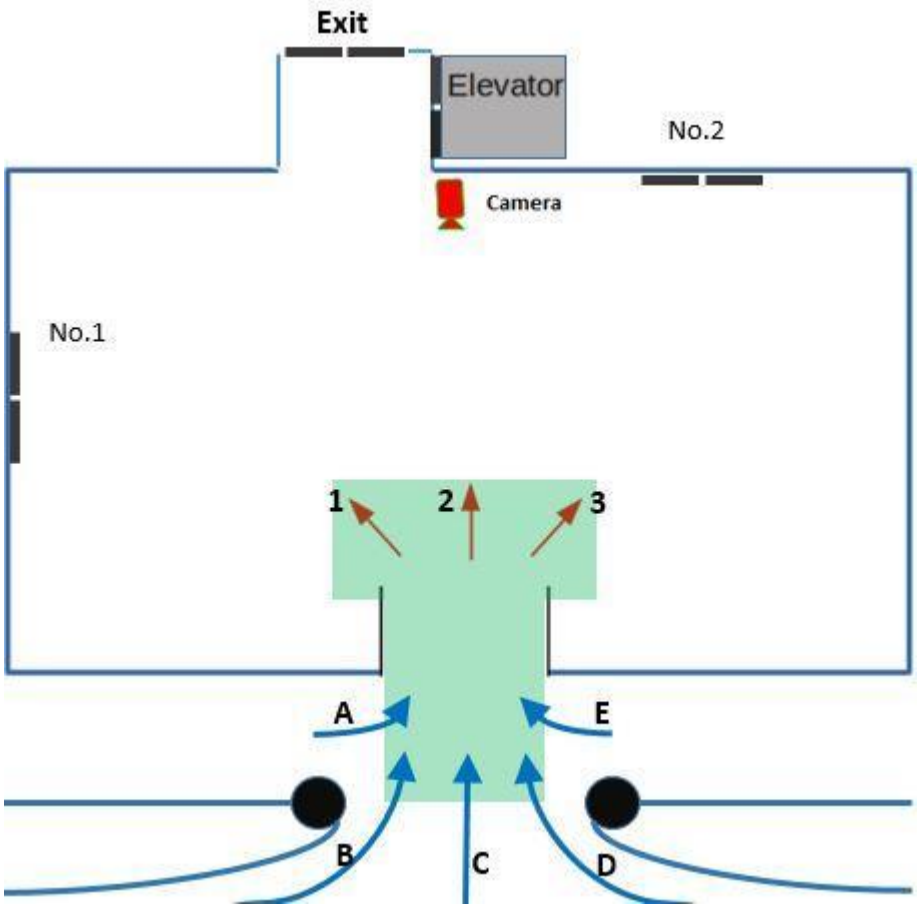


Figure 3.6 Data collection scenario.

The process of recording: when someone starts approaching the green area, start performing the recording until the person comes out of the frame of the camera (Figure 3.6). The reason we recommend to record in that area:

- Within that distance, the recorded image was achieved to a minimum size to facilitate feature extraction

- At that point, people's faces will tend to turn to the camera

Figure 3.7 Images recorded from the camera

Difficult cases can take place during recording:

- Many people come in at the same time, the frame will have more than one face. (Figure 3.7). The next step is to filter out facial images that belong to the same person.

- A person who is partially obscured by an object.

- One person moved quickly inside the hall. This may cause interference with the training. However, images that are not too blurred will be studied and trained later.

3.2.2.2 Labeling Dataset

Face images will be cut out from frames of videos by module face detection. These images are natural visual conditions in the lobby for several days and in real-life situations. Each staff's image has been labeled with their ID (Figure 3.8).

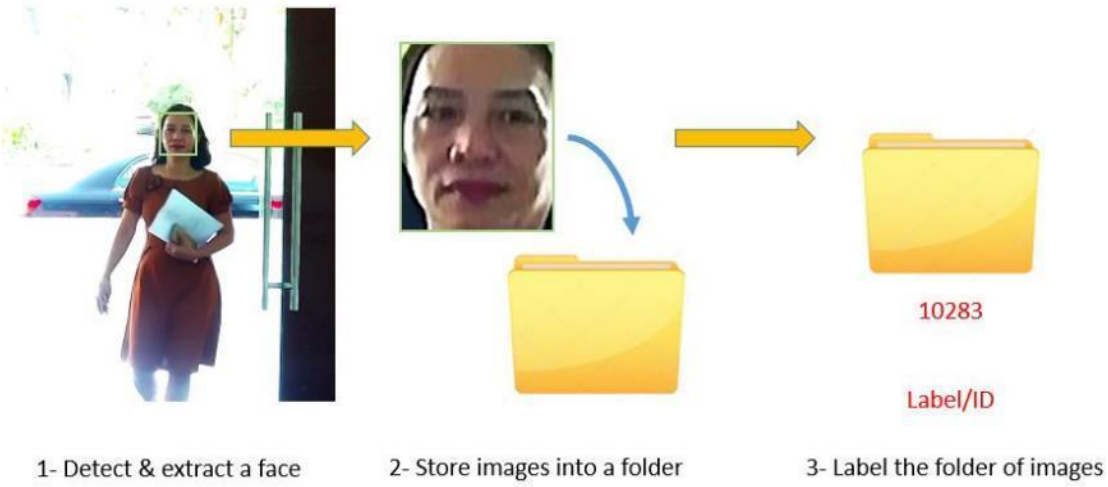


Figure 3.8 Image classification and labeling process

The images that are clear, have relatively good light, images with faces looking directly will be filtered out into a set called Set 1. This quality set is divided into two subsets: training set, testing set in well-matched condition (for short WM). Set 2 is the rest of the image and part of this set will be used for testing in very mismatched conditions (HM).

- Training set: includes face images that the face is directed towards the camera (which may deviate slightly to the left or right, 0 – 35 degree), image quality, and bright conditions are relatively good (the face could be seen clearly). The face is not obscured by other objects (masks, hats, etc.) or by the person walking ahead.

- WM set: Their contents and recording conditions are very similar to the ones for training. Sometimes the face shows a slight smile, most cases of lighting conditions are similar to the training set.

- HM set: There are many differences in operation and recording conditions when compared to the trained images such as: face blurred, face deviation angle from the main direction large (30 - 45), head down, eyes closed, hard lighting condition (too bright), smiling expression on face. This will be the test for the difficult cases encountered by the system. This dataset is similar to WM set in terms of quantity but differs in the properties of the face image.

General data are shown in Table 3.1 and the detail of the number of photos of each individual will be described in Appendix B.

Number of people	Train	Test		Sum
		WM	HM	
80	1561	661	10922	13144

Table 3.1 Description of the DUT dataset

In this dataset, there are many individuals with fairly little data (lack of 1 in situations such as: left, right tilt, straight view, brightness diversity). Moreover, the EM set also needs more data for testing. These data sets will be used to test the performance of ResNet - feature extraction and SVM. So the next part for purposes is to increase the amount of data and increase the diversity of the data set.

3.2.2.3 Data augmentation

This research needs many photos that contain enough facial features, so the data augmentation is used to increase face data to about 1,000 images for each

individual in the Self-built DUT dataset. The purpose is to automate image augmentation in order to expand datasets as input for machine learning algorithms, especially neural networks and deep learning.

The following are operations that are applied to expand datasets:

- Perspective skewing involves transforming the image so that it appears that you are looking at the image from a different angle. Skew or tilt an image either left, right, forwards, backwards (Figure 3.9) or by a random corner. The image will be skewed by a random amount in different directions.

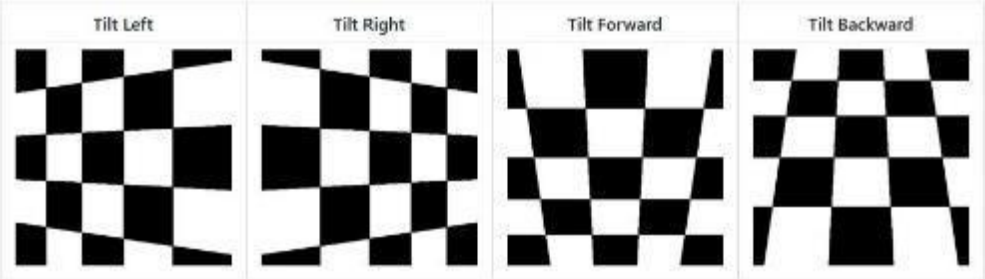


Figure 3.9 Four of the most common directions

- Elastic Distortions can make distortions to an image while maintaining the image's aspect ratio. It will be only used with a small margin to prevent the face from changing too much.

- Random Erasing is a technique used to make models robust to occlusion. We realized that some of the images after the face detection phase have the face partially obscured. Then we want to create data like that to increase the recognition rate in this case. (Figure 3.10)

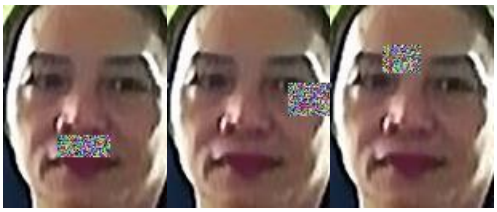


Figure 3.10 Example of random erasing on images

- Other operations are performed such as: rotating without crop, changing the brightness, contrast of the image. The brightness adjustment technique is used to expand the amount of data collected to the location (some places are dark; some places are brighter). However, the adjustment range will not be too large to accommodate the natural recording. (Figure 3.11)

Figure 3.11 Examples of brightness changes in face images

After data augmentation, we have a data set of 104000 images of 80 different individuals working in the building. The training set contains 700 images for each employee. The testing set includes: WM set and HM set. Both contain photos of 80 people with the number of testing images for each one is 300 images. The system will be trained with a training set and tested performance on the WM and HM set. This is called Test 4. Test 4 was performed with a feature extraction using ResNet (in section 2.2.1) and recognition with SVM (in section 2.2.2.3). The expanded data is shown in Table 3.2:

Number of people	Train	Test		Sum
		WM	HM	
80	56000	24000	24000	104000

Table 3.2 Description of the expanded dataset

3.2.3 The Self-built DUT_Emotion dataset

This is the emotional data set built from the DUT dataset. The purpose of this is to create a set of facial emotional data that is closely tied to the reality collected at the Smart Building. Because of the natural shooting, the number of emotions generated in the DUT dataset is few. Therefore, currently only 2 types of emotions are classified and used. The two types of emotions that have the most number of pictures are neutral face and happy face.

For each type of emotion, the number of images is also divided into 2 sets with 70% for training set and 30% for testing set. Detailed figures are described in Table 3.3.

Emotion	Train	Test	Sum
Neutral	91	39	130
Happy	77	33	110

Table 3.3 Description of the DUT_Emotion dataset.

Chapter conclusion: Through the chapter, these online datasets (UTK, AT&T, FERET and Extended Yale B) and self-built datasets (FASTData, DUT dataset) are introduced. Moreover, methods of constructing and analyzing the Self-built DUT dataset were presented. There are also operations to increase the amount of data, different situations under control. Training and testing sets split from the dataset are also considered. All of these datasets are used for performance testing of detection, facial recognition and emotional recognition modules.

CHAPTER 4 - System Setup and Implementation

Chapter introduction: In this chapter, our application will be described in detail. The design of the construction of the system is considered. Also, the web app has been developed in an elaborate way.

4.1. Proposal of the appropriate system

In order to meet the requirements of the university as the user unit, the team designed and implemented an integrated system for managing people in and out of the building based on facial recognition technology as the diagram in Figure 4.1.

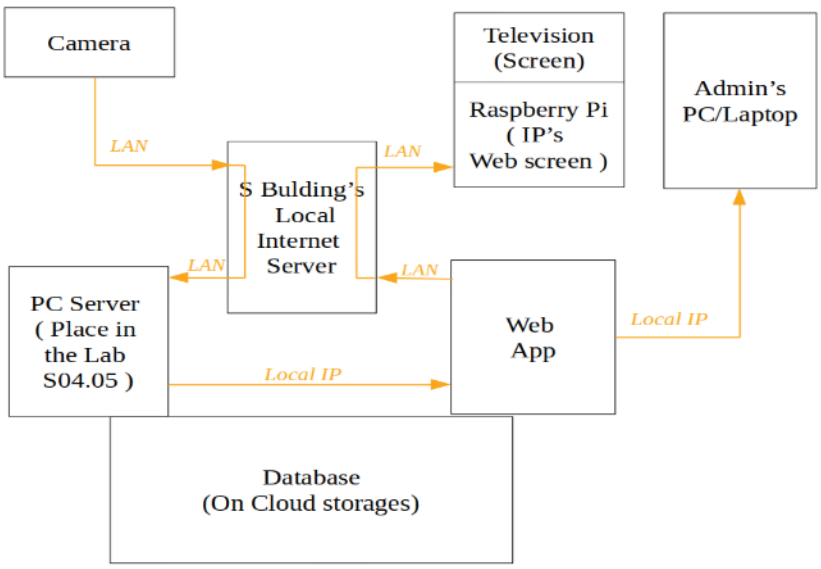


Figure 4.1 Block diagram connecting devices in hardware systems

The system includes video recording, transmission, identification processing software, information display software on the welcome interface, and software that provides monitoring information for authorized personnel to use. Attached the statistical data. Specifically, the system is designed to consist of main blocks with the following functions:

- Hardware:
    - Camera records, capable of handling glare, exposure compensation and zoom control via software
    - A high-speed computer handles the training of large numbers of samples and performs identification.

- The screen displays welcome information, connects wirelessly to the system via an embedded computer (to connect to the remote server system)
- Employee personal computers are authorized to access management applications.
- Local Internet - LAN connects the above devices.
- Parameters of equipment in the system and setup the system are described in detail in Appendix A

- Software:
  - User identification software for machine learning techniques
  - The web application welcomes Staff to work with the information displayed
  - Web application that provides management information

The operating procedure of the system is designed as follows:

- The camera records and stores information as frames into memory
- The facial recognition algorithm on the server computer will perform the identification and return information as the ID of the identified individual.
- The ID-based Web App management software will access the database of school personnel information to provide personally identifiable information, then display the information on the welcome screen.
- At the same time, personal related data that has been identified or notified as "strangers" will be stored on the school's server and OneDrive for convenient access to personal information to the building, serving the management.

4.2 Overview of Web application

Welcome web application and employee management are developed, integrated with the system. Applications and interfaces are written in many languages JS, CSS, HTML, Python, PHP, MySQL, Shell script and MQTT method, HTTP/HTTPS protocol.

MQTT stands for **M**essage **Q**ueuing **T**elemetry **T**ransport - There are a few basic concepts (Figure 4.2):

- Publish/Subscribe
- Messages
- Topics
- Broker

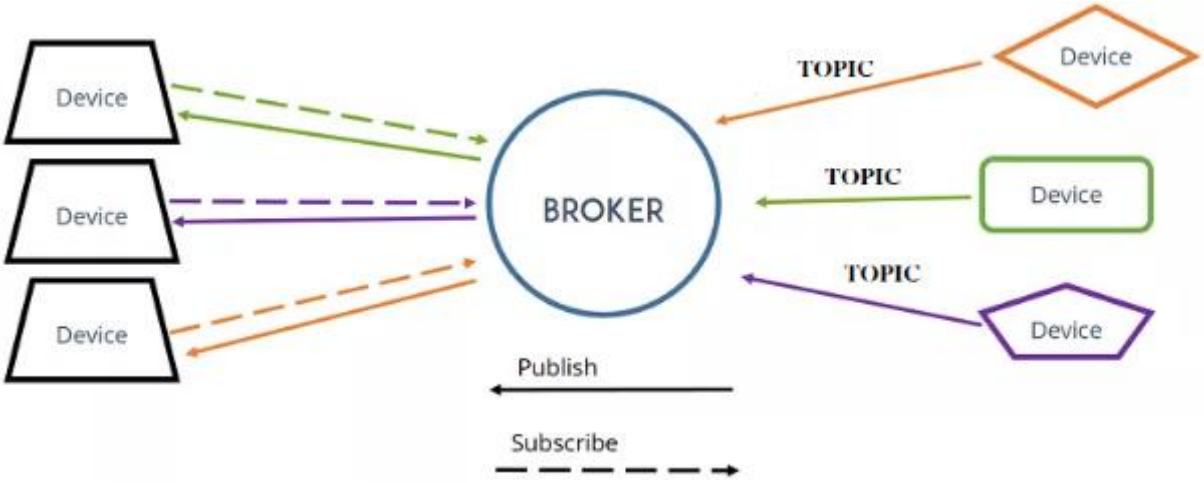


Figure 4.2 Overview MQTT - Broker

The **broker** is primarily responsible for receiving all messages, filtering the messages, deciding who is interested in them and then publishing the message to all subscribed clients.

The **Toptics** - Another important concept are the topics. Topics are the way you register interest for incoming messages or how you specify where you want to publish the message.

Topics are represented with strings separated by a forward slash. Each forward slash indicates a topic level.

And in the project we use "/ids" topics with port 1883 and server: "localhost", forwards information/data performance and real-time responsibility. Special to save memory execute systems.

In front - end of Web App, we used HTML, CSS, JS to create Web App about Back-end we used to Python, php together combined to create the best result. And we used MySQL to build the database. The end, we used a Shell script to run and set up all programs, it's the function to compile and allow access to the system in the best way.

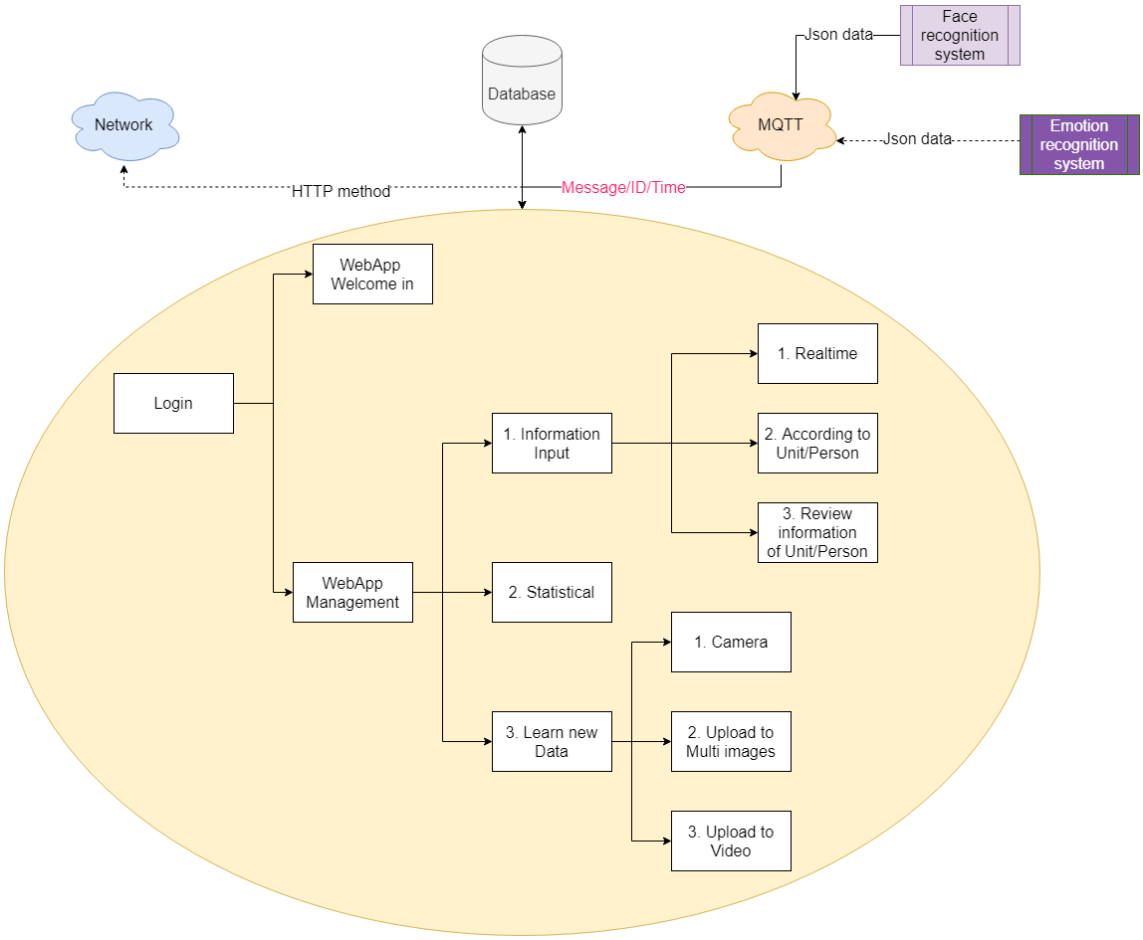


Figure 4.3 System diagram structure.

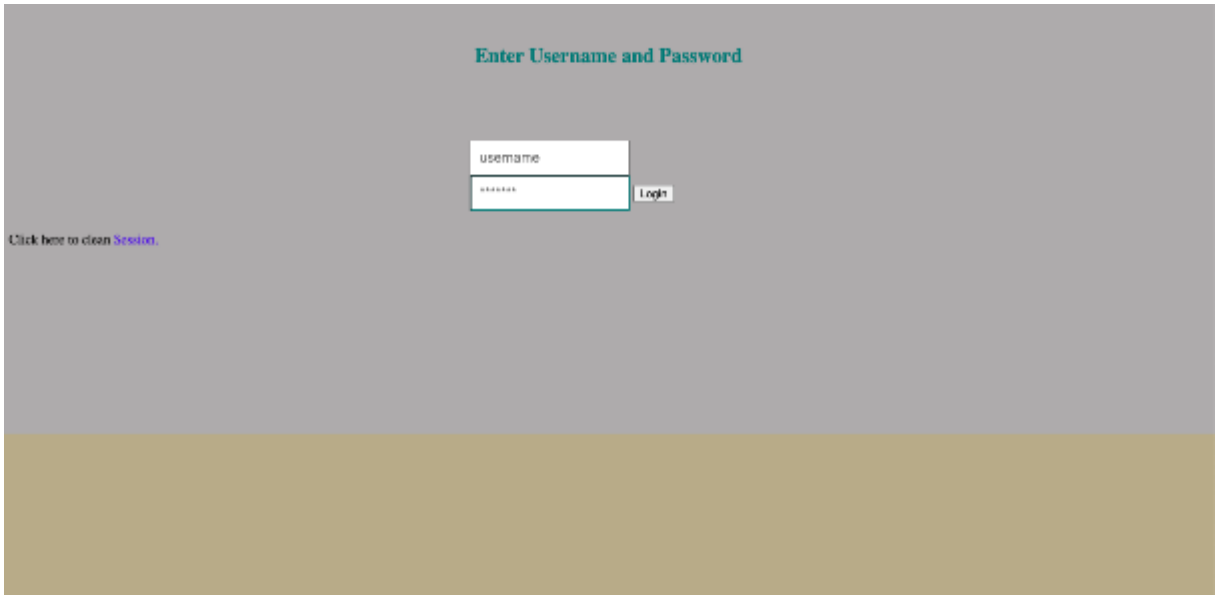


Figure 4.4 Login - system Web App.

4.3 Welcoming application

Welcome interface with a neat layout, streamlined function frames to attract the attention of people entering the building. In addition, the welcome interface has the function to change the wallpaper as well as cute images to create a new and interesting feeling for the viewers. With this design, the management system does not create the feeling of being monitored, but on the contrary, it is very exciting to be welcomed and cared for.

The features:

- Security features login, logout.
- For individuals working in administrative building: information on their full names, emotions, positions and units will display on screen as shown in Figure 4.5.
- For individuals who do not work at a building or a stranger: general greetings as shown in Figure 4.6.
- Information about the event will take place at the building and related participants.
- Other background information: time, dates, and major holidays throughout the year.

For many people entering the building, the screen randomly greets 1 person in the frame for 2 seconds, then continues randomly as they leave the frame. Future updates of the module: position, higher rank will take precedence - If the position is equal, then will greet randomly.

Figure 4.5 Welcoming screen for a person in the database.

Figure 4.6 Welcoming screen for stranger or not having a person in frame.

4.4 Web application for management

The managed web application provides the following information:

- Security features login, logout.
- All information (including name, emotion, unit, position, image) of an individual entering the building in chronological order and tracking images at the time that person enters the building.
- Statistics of staff frequency working in the building by the chart.
- Features import and train new people.

Login Web App - Management:

Figure 4.7 Home page

Form login account admin of manager:

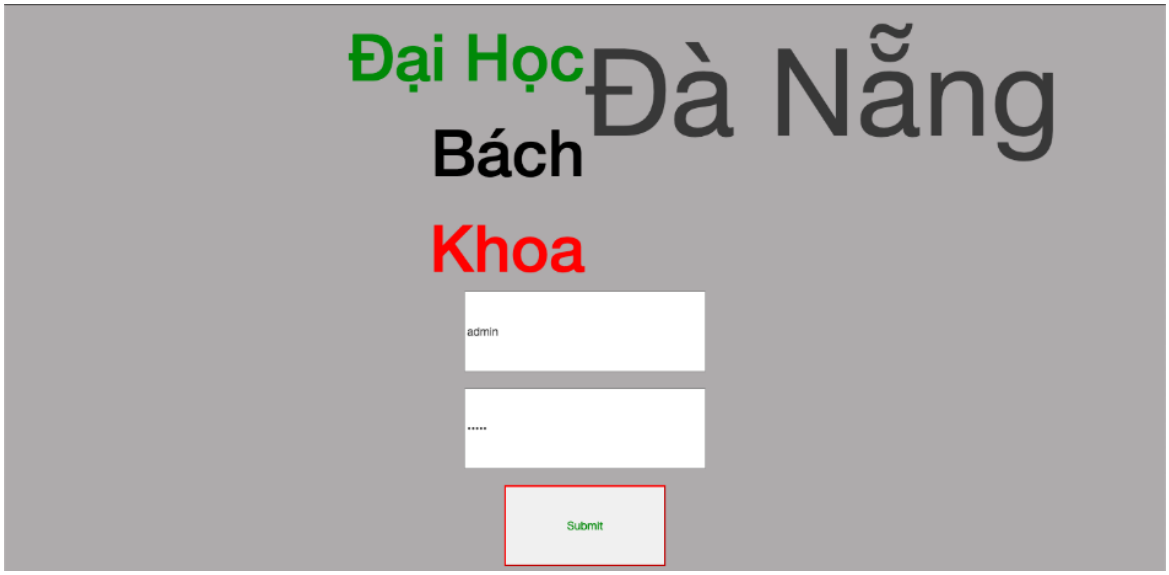


Figure 4.8. Form login account admin of manager

Details of Web App - management about many main functions:

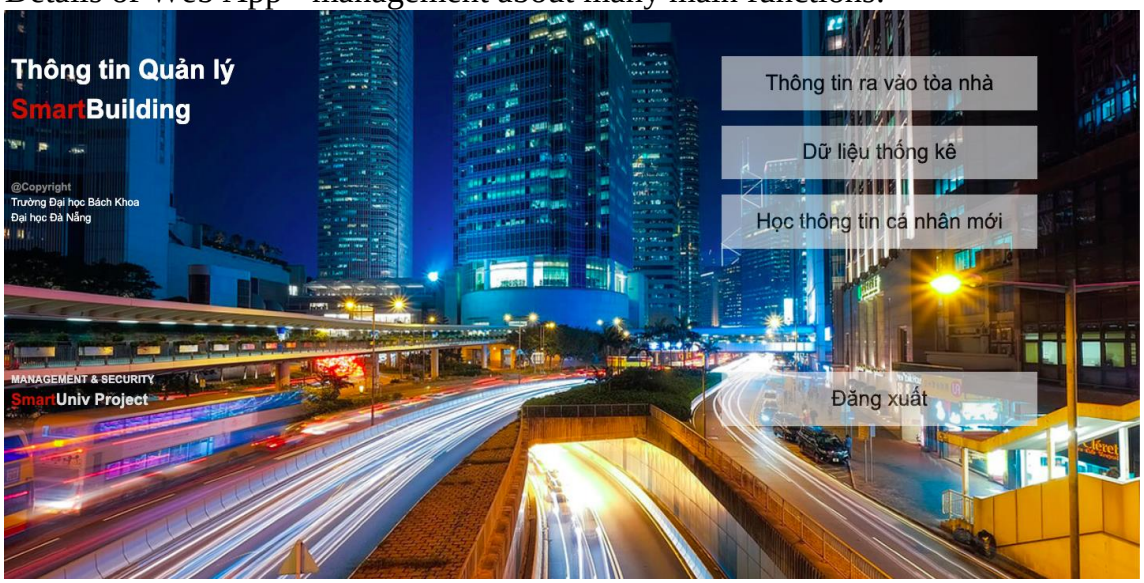


Figure. 4.9. Management page.

Web App management has 3 options. Information about persons entering the building will be accessible via the first option. Currently, the system can access by time, by unit, by specific individual name as shown in Figure 4.10. Details of the options will be presented in Appendix B.

Figure. 4.10. Options information view follows on the first button.

The second option is to view the graph of the statistics of people entering the building by each unit. (Figure 4.11). The SQL (Structured Query Language) internal join method is used for compiling and getting data from both Real-time table and Manager Table Database. The system analyzes and calculates according to the calculation formula (Unit: Year):

A% = (Numbers 1 Appear Unit)/ (Total appear of personal of all units) *100% (3)

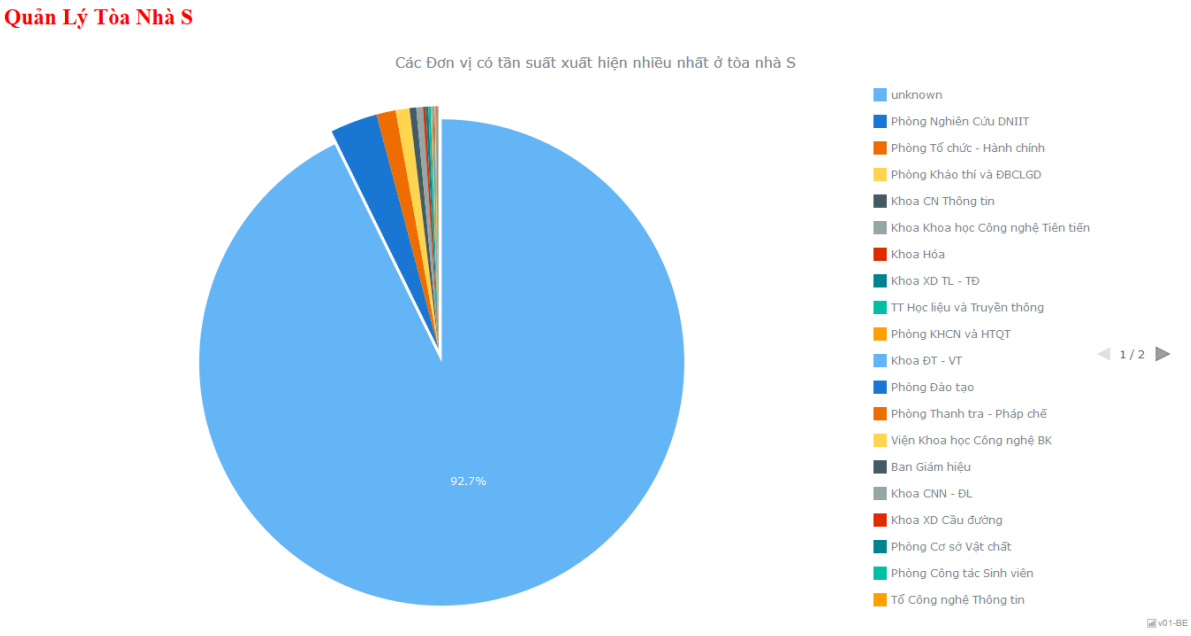


Figure. 4.11. The Statistical (PIE Graph).

The last option is the function to learn new people for the system data. This can be done in three ways, as shown in Figure 4.12. The process for adding new individuals to the system is discussed in detail in Appendix B.

Figure 4.12. Options for learning new people.

After uploading data for new people, the system must be restarted. In Figure 4.13, Button 1 will restart the system immediately, button 2 will restart the system at 0:00.

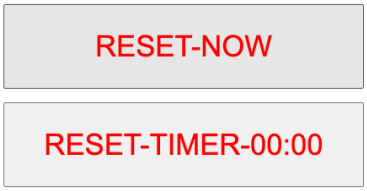


Figure 4.13. Reset the system.

Chapter conclusion: Through the chapter, the steps for setting up the system were covered in this chapter. The flow of the application and how the application manages operations has been presented.

CHAPTER 5 - Experimental Results and Evaluation

Chapter introduction: In this chapter, the method for face detection, face recognition, face emotion recognition are evaluated performance based on certain criteria and discuss the evaluation result.

5.1 Criteria

In order to evaluate the performance of the proposed methods, the following metrics are used: confusion matrix, precision, recall, accuracy, F1-Score:

- Confusion Matrix is a method of describing the effectiveness of a classification model, including a table with vertical axis as true label and horizontal axis as predictive label. Each cell in the table contains the number of times the result was found in that case. The advantage of using a confusing matrix is a visual representation of how well or not it works for each class. In this study, classification measures were used to test the effectiveness of the proposed facial recognition model.

- Recall(RE) is the ratio of the total number of positive predictions to precise predictions:

$$RE = \frac{TP}{TP + FN} \quad (4)$$

- Precision(PRE) is the ratio of correct positive predictions to the total positive predictions:

$$PRE = \frac{TP}{TP + FP} \quad (5)$$

- Accuracy(ACC) describes how often the model predicts correctly:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

- F1-Score(F1) is calculated from accuracy and retrieval, which is used to find the balance between the two values

$$F1 = \frac{2 * PRE * RE}{PRE + RE} \quad (7)$$

TP: is an outcome where the model correctly predicts the positive class;
FP: is an outcome where the model incorrectly predicts the positive class;
FN: is an outcome where the model incorrectly predicts the negative class;

TN: is an outcome where the model correctly predicts the negative class;

5.2 Result for face detection

Testing the examined algorithms on The UTK face dataset [24]. The FaceBoxes method gives the best result as shown in Table 5.1 below.

	Haar Cascade	HOG + SVM	CNN	MTCNN	FaceBoxes
Front face	100	100	100	100	100
Face with glasses	0	95	100	100	100
Non-front face	0	35	98	100	100
Covered face	0	0	29	98	98
Recall	59.40	65.11	67.76	93.85	95.67
Precision	63.03	73.10	96.25	97.02	98.56
F1-score	61.16	68.87	79.53	95.40	97.09

Table 5.1 Performance of Face Detections (%)

According to the result, The Haar-cascade method can detect faces at different scales, but its major drawback is that it does not work on non-frontal images and under occlusion. The HOG and SVM can work under most cases except small faces, side faces and extreme non-frontal faces like looking up or down, and sometimes. The CNN can detect for different face orientations. The MTCNN is the better accurate algorithm with F1-score equals 95.40%. The most accurate algorithm is FaceBoxes with F1-score equals 97.09%. It can detect faces with faces at different orientations – up, down, left, right, side-face, etc., and it can work even under glasses or covered faces.

Then, we test the processing speed of three detection packages and the FaceBoxes to choose the best method that suits the system. Three detection packages using MTCNN will be considered in section 2.1. Each method is tested for its speed in detecting the faces in a set of 300 images (300 frames from one of 1-minute video recorded), with GPU support enabled. Detection is performed at 3 different resolutions. The computational time for each single face in a frame is given in Table 5.2.

Computational time (s)	Dlib	Keras	Facenet-pytorch (batched)	FaceBoxes
(540x960)	0.05	0.16	0.04	0.02
(720x1280)	0.09	0.26	0.07	0.05
(1080x1920)	0.19	0.50	0.12	0.09

Table 5.2 The computational time of four face detections

In terms of the calculation time of face detection methods, FaceBoxes is the fastest algorithm with all 3 resolutions, followed by Facenet-pytorch (batched), Dlib and finally Keras is the lowest method. It is also easy to see that if we resize the camera resolution to (540x960), the face is still well detected with about 3 times faster calculation time.

So, with the face detection module, we decided to use the FaceBoxes method combined with resizing the video resolution from the camera to a resolution that was reduced by about 3 times compared to when direct monitoring from IP camera.

5.3 Result for face recognition

Test 1, Test 2, Test 3 were performed with a feature extraction using ResNet (in section 2.2.1) and recognition with Euclidean distance (in section 2.2.2.1). Test 4 was performed with a feature extraction using ResNet (in section 2.2.1) and recognition with SVM (in section 2.2.2.3).

5.3.1 Test 1: A large number of individuals

The model was tested with increasing numbers: 50, 500 and 1,000 different faces. The number of people influencing the result, the identification rate was reduced by about 4% with the number of people being 1000, compared to testing on 50 faces. As can be seen, the face recognition model has also achieved relatively good efficiency with over 90% of parameters. Details are shown in Table 5.3.

FERET Dataset			
Number of people	**50**	**500**	**1000**
RECALL	94,63%	92,11%	90,60%
PRE	96,82%	94,90%	94,24%
ACC	94,63%	92,11%	90,60%
F1 SCORE	95,00%	92,30%	91,05%

Table 5.3 Evaluation results according to the number of people of the FERET.

5.3.2 Test 2: Different lighting situations

In the Extended Yale B dataset, the light source is distributed on a hemisphere with the angle of illumination described in Figure 5.1, including the azimuth and the height of 64 light bulbs. Each point represents the location at which the light bulb entered.

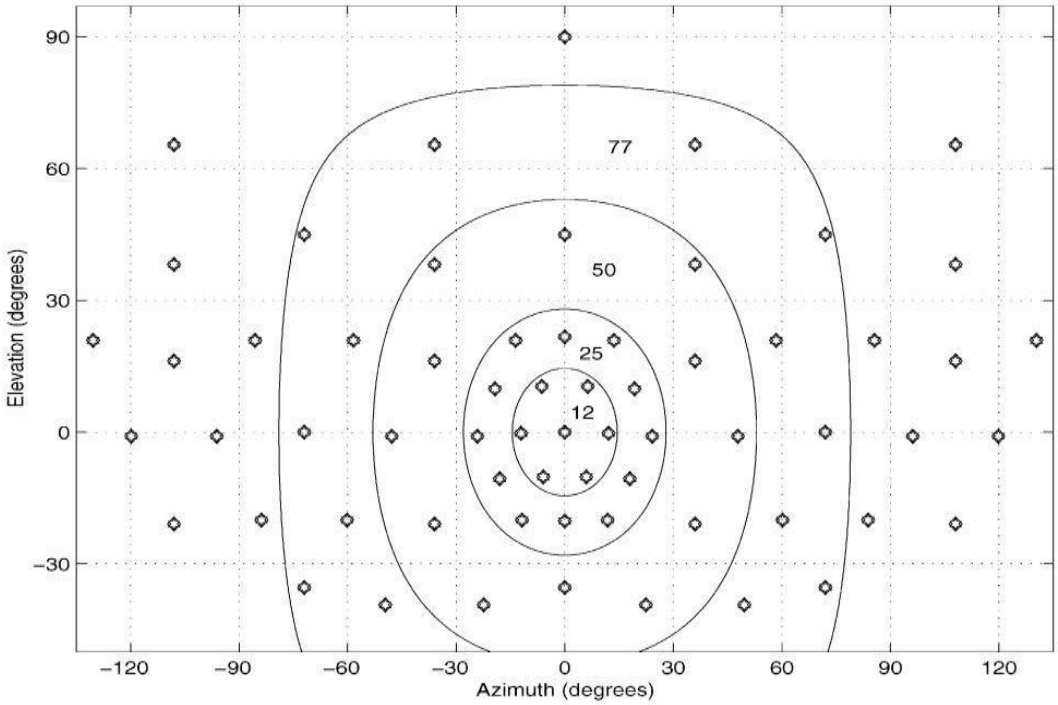


Figure 5.1. Diagram of the position where the light source is located.

Test results 2 on different light angles are shown in Table 5.4.

Extended Yale B dataset				
Lighting angle	**0-25º**	**25-50º**	**50-77º**	**77-130º**
RECALL	100,00%	99,04%	98,31%	85,71%
PRE	100,00%	99,11%	98,49%	83,33%
ACC	100,00%	99,04%	98,31%	85,71%
F1 SCORE	100,00%	99,02%	98,25%	84,13%

Table 5.4 Results of evaluation according to different lighting angles of Extended Yale B.

5.3.3 Test 3: Different positions of the face

The results achieved for the evaluation criteria: RE, PRE, ACC, F1 are described in Figure 5.2.

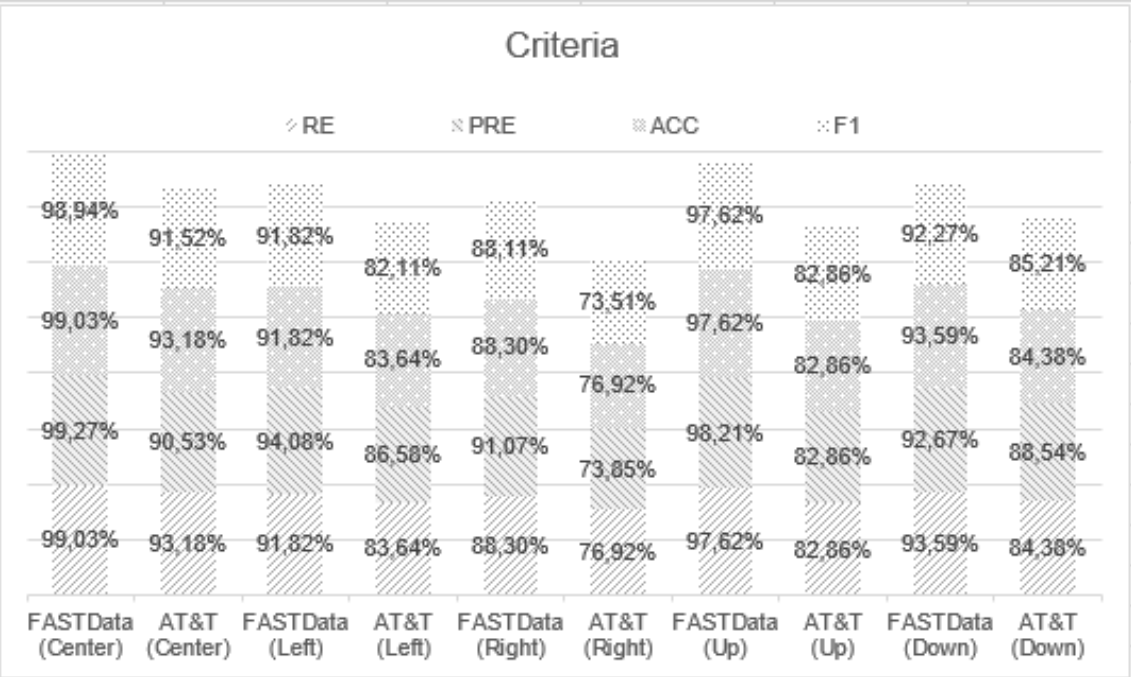


Figure 5.2. The results of RE, PRE, ACC, F1 criteria are achieved with AT&T and FASTData data sets respectively.

With AT&T database: The result shows that the model can detect very well different positions of the face: Face Center, Half Left, Half Right, Head Up, Head Down as described in Figure 5.3. However, in the case of Half Left and Head Down, the result is lower.

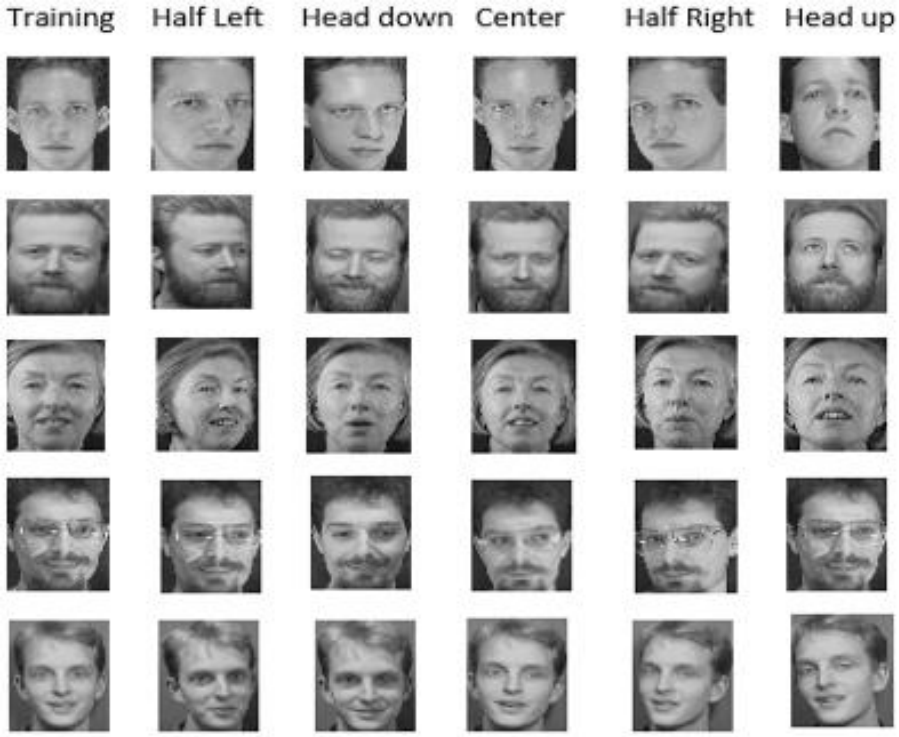


Figure 5.3. Image matrix for training and testing on AT&T dataset.

With FASTData dataset: there are relatively high results and in different positions of the face, the results are only reduced by about 1-5%. The case of Head Up is high due to the sample obtained is similar in characteristics to the face model used for training. Better image quality from the digital camera also makes this result better. It is described in Figure 5.4.

Figure 5.4. Image matrix for training and testing on FASTData dataset.

5.3.4 Test 4: Performance on the Self-built DUT dataset

The performance of trained model with SVM is shown in Table 5.5.

Kernel	Regularization Parameter (C)	Train time (s)	Recognition time (ms)	F1-score (%)	
				WM (*)	HM (*)
Linear	0.01	3555	13.33	91.50	72.92
	1	205	5.10	98.99	86.17
	100	91	2.44	99.13	85.43
RBF	0.01	4104	13.89	91.14	73.18
	1	213	5.43	99.23	85.82
	100	110	2.73	99.18	85.75

Table 5.5 Evaluation result of SVM model with different parameters

According to the result shown in the table, we decided to use the RBF kernel with C = 100 to compare with other methods. For now, when mentioning the SVM model without specific parameters, it means we are mentioning this model.

After training on the training set, we evaluated performance of the SVM model by testing on testing set: WM set and HM set.

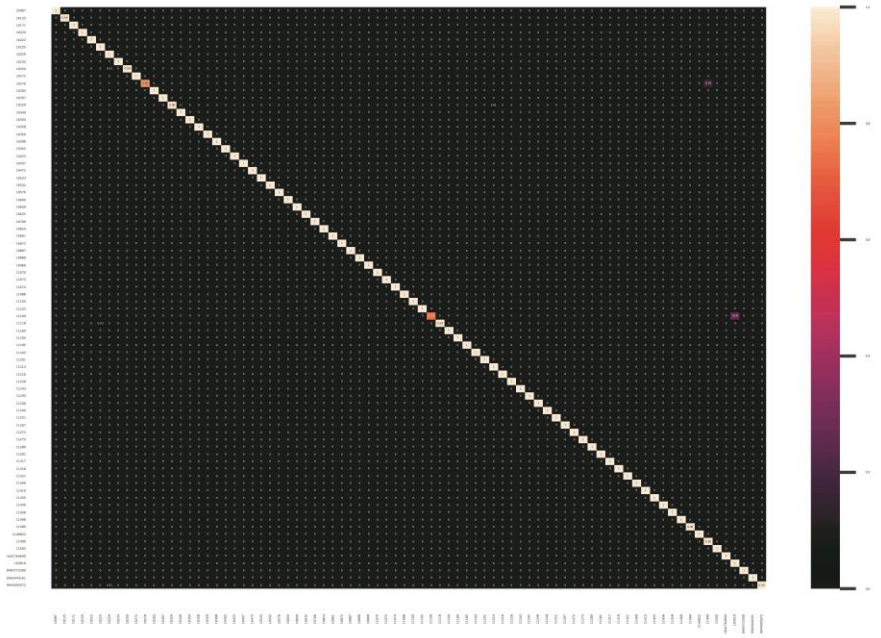


Figure 5.5 Confusion matrix on WM set.

As shown in Figure 5.5, most of the results were highly accurate, with only two individuals reporting mistaken identity to another person. A review of the data showed that the two cases had nearly the same faces and wore glasses.

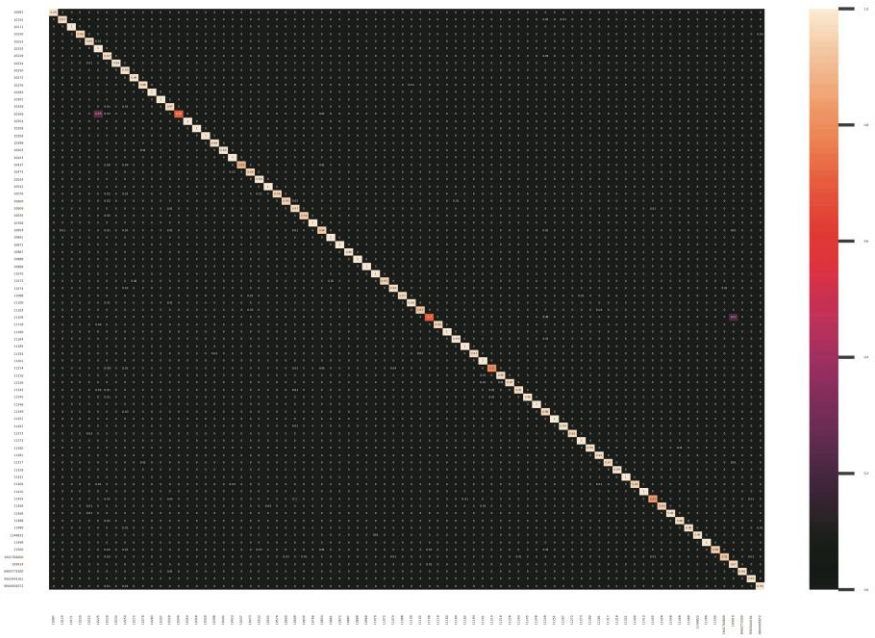


Figure 5.6. Confusion matrix on HM set.

In Figure 5.6, more cases of false identification began to appear. Most of these cases have faces that are nearly equivalent to the angled angles of the face and have glasses.

Test results with WM show that when the system is trained enough cases, the ability to identify correctly will be very high. In the HM test set, it contains different images than training such as: worse conditions (brighter or darker), blurred images, faces turning to the left (or right) at large angles, smiling faces. Test result is shown on Table 5.6.

Scenario	Algorithm	PRE	RE	ACC	F1
WM	Euclidean	90.24	89.65	89.65	89.94
	KNN	97.59	97.38	97.38	97.48
	SVM	98.71	98.55	98.55	98.62
HM	Euclidean	71.91	67.71	67.71	69.74
	KNN	80.90	78.85	78.85	79.86
	SVM	85.50	82.97	82.97	84.21

Table 5.6 Results of evaluation of the Self-built DUT dataset (%)

According to Table 5.6, on a well-matched testing set, three algorithms all give high evaluation scores, over 90% of prediction is correct. Moreover, the KNN and SVM model give 9-10% accuracy gain in comparison to the model that uses Euclidean distance. This proves that KNN and SVM is more reliable but the Euclidean distance model has an acceptable accuracy. Although with the SVM model, there are 1.5% of miss-classification. Reference to the confusion matrix, this error mainly comes from only 2 individuals that the model can't distinguish between them.

On highly-mismatched testing set, there is a slight reduction in performance of the Euclidean distance model, 22% lower than the result of the well-matched testing set. The reason is the model trained with Euclidean distance doesn't generalize well so with data that differ from the training set, the model does not work well. Besides, the KNN and SVM maintain good performance on this testing set. The accuracy's difference between KNN and SVM is around 4%, more than the difference in the well-matched testing set which is around 1%.

The recognition time for each single face in a frame is given in Table 5.7. This

is the average value when we test the models with WM and HM testing set.

Euclidean	KNN	SVM
2.4ms	16.1ms	11.3ms

Table 5.7 Computational time of face recognition

Recognition time of a single face (unit: ms). Evaluation results are evaluated on the configuration of core i5-9300H laptop, single core mode, 8.00 GB RAM, ubuntu 20.04.

5.4 Result for emotion recognition

It is a model tested on DUT_Emotion dataset sets, the model focuses on these two weights (happy, neutral). The results of the emotional recognition module are shown inFigure 5.7)

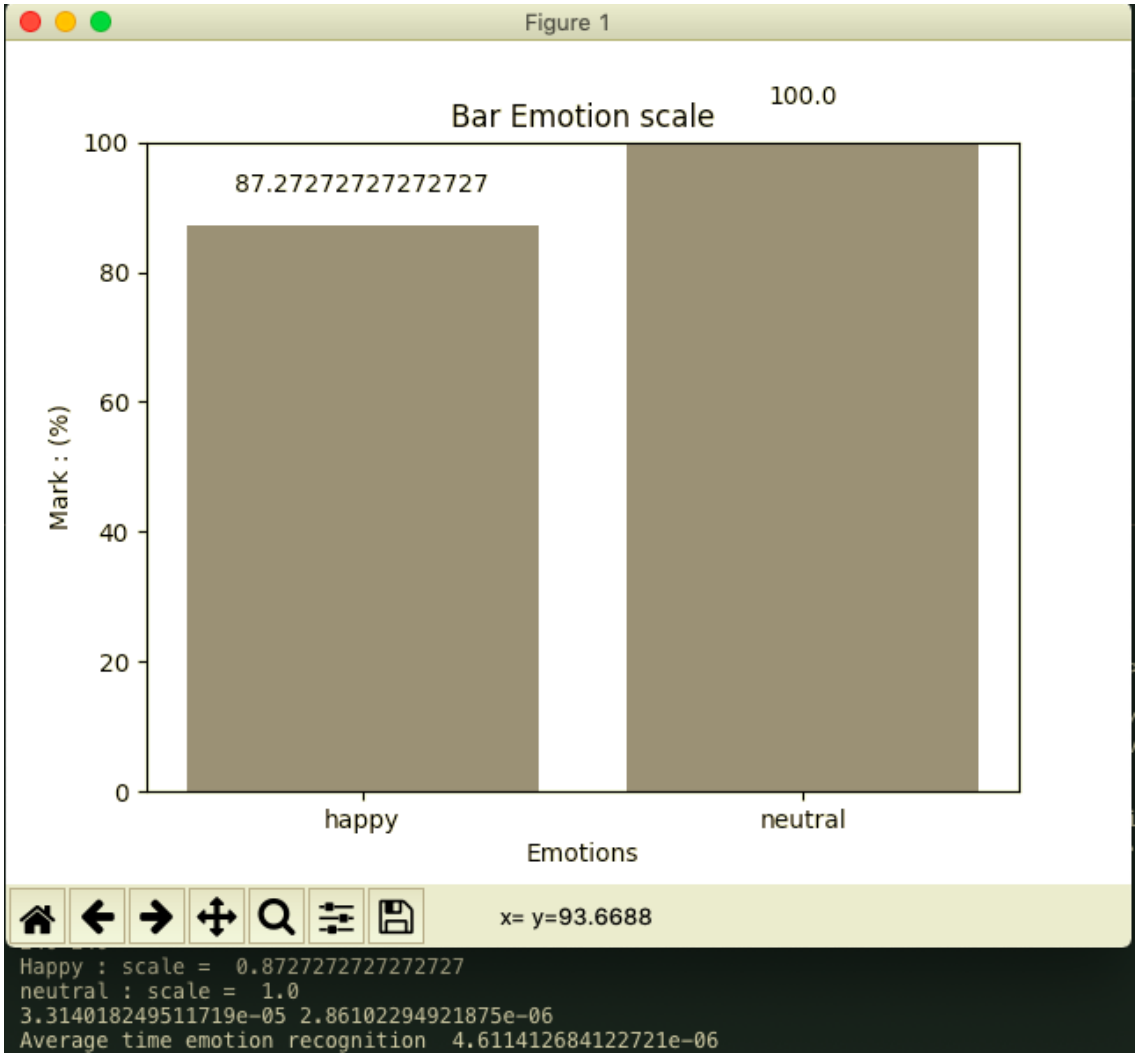


Figure 5.7. Result on DUT_Emotion dataset.

Mini-Xception model was pre-trained on the DUT_Emotion dataset. About Figure 5.7, the accuracy rate is good. Therefore, the data lacking diversity also does not achieve the ratio too well.

Findings:

Emotions with strong facial expressions or special facial characteristics such as happiness and neutrality often achieve good results with greater accuracy and recall. In labeling and classifying emotions, it is difficult to classify with 100% certainty.

The recognition time for each single face in a frame is given in Table 5.8. This is the average value when we test the models with the testing set.

Max-time	Min-time	Average-time
3.50×10^{-5}s	2.62×10^{-6}s	4.52×10^{-6}s

Table 5.8 Computational time of emotion recognition

5.5 System evaluation results

5.5.1 System test in realtime and real condition

To test the performance of all the modules and to check if the communication between the modules is correct, we give a real video stream from the camera at the S-Building hall to the system and we check the output shown on the Web application. We run the test on 1 day and record the input video to validate with the system output. In the morning, from the camera's perspective, the building is very dark inside, and the outside has objects from the construction that reflect the light directly into the camera. This makes the face quality from the camera significantly reduced and it is impossible to detect faces in such a condition. Currently, this problem cannot be fixed immediately. Therefore, we only take the test results in the afternoon time frame from 12pm to 6pm. The summary of the test will be shown in Table 5.9.

Time	ID	Scenario	In Dataset	Image quality	Result
12:06:28	10542	Go straight	Yes	glare in the middle	Correct
13:18:14	10272	Go straight, then turn right	Yes	glare in the middle	Correct
13:21:28	11249		Yes	glare in the middle	Correct

13:40:40	11455		Yes	glare in the middle	Unknown
13:42:51	11193		Yes	glare in the middle	Unknown
13:46:05			No	glare in the middle	
13:46:36	11185	Go straight and bow	Yes	glare in the middle	Unknown
13:49:07	11458	Wear a mask	Yes	glare in the middle	Unknown
13:50:31	11100	Wear a mask, then take it off	Yes	glare in the middle	Unknown
13:51:43		Wear a mask	Unknown	glare in the middle	
13:53:59		Wear a mask	Unknown	glare in the middle	
13:56:41		Wear a mask	Unknown	glare in the middle	
14:03:08		Wear a mask	No	glare in the middle	
14:05:00		Go straight	No	glare in the middle	
14:06:24		Go straight and bow	Unknown	glare in the middle	
14:13:41	10329	Go straight	Yes	glare in the middle	Correct
14:14:21			Unknown	glare in the middle	
14:17:30		Start from the left, enter the door and turn right	Unknown	glare in the middle	
14:41:02		Start from the left, enter the door and turn right	No	glare in the middle	
14:41:17	11272	Go straight then turn right	Yes	glare in the middle	Unknown
14:42:00	10609		Yes	glare in the middle	Unknown
14:46:55			No	glare in the middle	
14:59:11	11499		Yes	glare in the middle	Correct
15:01:32	11193	Wear a mask	Yes	glare in the middle	Unknown
15:06:02			No		
15:12:03			No		
15:13:33	11102	Go straight and bow	Yes		Unknown
15:14:14		Go straight and bow	No		
15:22:57		Wear a mask	Unknown		
15:31:20	11074		Yes		Correct
15:43:24			No		
15:45:28	11280	Go straight, look at the screen	Yes		Unknown
15:46:58		Go straight	No		

15:47:47		Face tilted to the left	No		
15:49:21			No		
15:52:12		Go straight and bow, wear glasses	No		
16:00:30	10307		Yes		Correct
16:12:51	10609		Yes		Unknown
16:14:28			No		Mistake (11455)
16:28:33		Go straight	No		
16:24:21	11074		Yes		Correct
16:53:30	11226		Yes		Correct
17:02:36	10542	Go straight	Yes		Correct
17:04:27		Start from the left, enter the door and turn right	No		
18:02:10	10473		Yes		Correct
18:02:42	10542		Yes		Correct

Table 5.9 Testing in real-time

In the afternoon, the light reflected straight into the camera decreased a bit. Therefore, the actual test results seem better. According to Table 5.9, various situations appear during this period of time (go straight, turn left, bow, wear glasses, wear a mask, …). In addition, there are strangers and people who have images in the DUT dataset. Calculating the results achieved for those who have in the system data, we get the correct identification rate on the overall is 12/23. Therefore, it can be said that the accuracy rate of the system in real time at this period of time is about 52.17%. This figure is temporarily acceptable due to inclement lighting conditions and incomplete databases.

5.5.2 The processing time of the system

The processing time of each main module and the total processing time of the whole system are shown in Figure 5.8. These results have been obtained by testing the system on the following hardware computer: 4.1 GHz core i5-9400F, GPU mode with 16.00 GB RAM, 6GB VRAM GPU NVIDIA RTX2060.

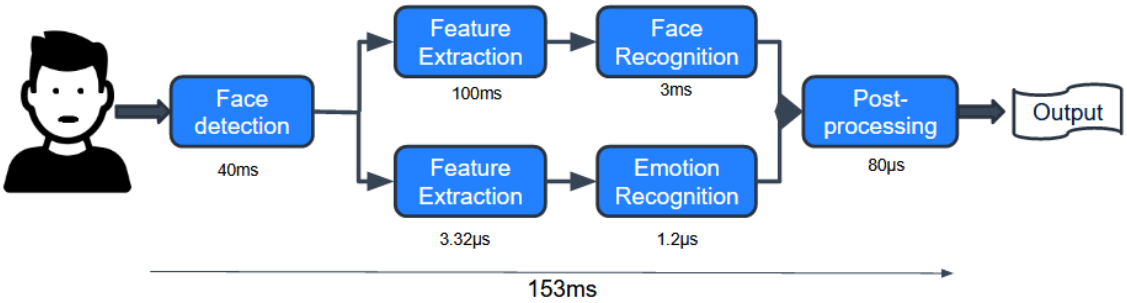


Figure 5.8. Computational time

Chapter conclusion: Through the chapter, test results were presented along with the facial recognition module and processing time. In addition, the results of the emotional recognition assessment are also presented.

CONCLUSION

Face detectors have been tested on various evaluation criteria, from which we will have references on which method will apply to our system, as well as the need and purpose used for future development systems. The experimental results show that the performance of the proposed face detection technique which uses the FaceBoxes pre-trained model as a face detector on both normal and small face images, leads to quite fast and robust for face detection module, meets the real-time system requirements.

With face recognition module, following the evaluation result, we compared many methods that can be applied for face recognition. Because of the high requirements of accuracy and recognition time, the SVM model trained on ResNet face feature extraction is the appropriate method for our problem. On a well-matched testing set the recognition accuracy reaches more than 98% and on high-mismatched the accuracy is lower, more than 82%.

With the emotion recognition module, following the evaluation result. We have proposed and tested a general building design for creating mini-Xception. Our proposed architectures have been systematically built in order to reduce the amount of parameters. We began by eliminating completely the fully connected layers and by reducing the amount of parameters in the remaining convolutional layers via depth-wise separable convolutions. On a dataset testing set the recognition accuracy reaches more than 87.27%, 100% for 2 states in turn to be happy, and neutral.

Successful applied research: an integrated system of human face recognition software that communicates with cameras and information displays installed at the administrative building at The University of Danang, University of Science and Technology. The software includes a welcome interface module and a web-based information management interface module.

This proposed system will be always upgraded and improved to achieve the best performance and further developed with many different purposes according to user needs.

PUBLICATIONS

For a long time of system research and development, our team has been greatly guided by our Supervisor council including Prof. Nguyen Thi Anh Thu and Prof. Pham Van Tuan in publishing articles at several national conferences. Information about these conferences is shown below:

- The 8th CITA 2019 *(The 8th Conference on Information Technology and its Applications);*

- The 22st REV-ECIT 2019 *(The 22st National Conference on Electronics, Communications and Information Technology);*

- The 5th ATIGB 2019 *(The 5th National Scientific Conference On Applying New Technology in Green Buildings);*

Prizes that this project has won at scientific research students:

- DUT Promising Prize at Smart Campus University of Danang 2018-2019

- FAST TECH TOP at FAST Techshow 2019

- Third Prize at FAST Techshow 2019

- Second Prize Ideas at BK Techshow 2019

- First Prize at FAST Techshow 2020

- First Prize at BK Techshow 2020

BIBLIOGRAPHY

[1] Shilpi Singh, S.V.A.V, "Prasad, Techniques and Challenges of Face Recognition", in *A Critical Review, Procedia Computer Science 2018,* Volume 143, 2018, Pages 536-543, ISSN 1877-0509.

[2] Juneja, Komal, "An improvement on face recognition rate using local tetra patterns with support vector machine under varying illumination conditions", in *IEEE Computing. 2015 International Conference on Communication & Automation (ICCCA),* India, pp. 1079 - 1084, May 2015.

[3] Jia Jun Zhang, Yu Ting Shi, "Face recognition systems based on independent component analysis and support vector machine", in *IEEE 2014 International Conference on Audio, Language and Image Processing (ICALIP), Shanghai*, pp. 296 – 300, July 2014.

[4] G. Majumder, M. K. Bhowmik, "Gabor-Fast ICA Feature Extraction for Thermal Face Recognition Using Linear Kernel Support Vector Machine", in *IEEE 2015 International Conference on Computational Intelligence and Networks (CINE), Bhubaneshwar*, pp.21-25, Jan. 2015.

[5] P. Viola and M. J. Jones, "Robust real-time face detection," in *International Journal of Computer Vision*, vol. 57. no. 2. pp. 137-154. 2004.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, IEEE Computer Society Conference 2005*. vol. 1. pp. 886-893.

[7] H. S. Dadi, G. K. M. Pillutla, "Improved Face Recognition Rate Using HOG Features and SVM Classifier", in *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)*, Volume 11, Issue 4, Ver. I, pp. 34-44, Jul-Aug. 2016.

[8] Nawaf Hazim Barnouti, "Improve Face Recognition Rate Using Different Image Pre-Processing Techniques", in *American Journal of Engineering Research (AJER)*, Volume 5, Issue 4, pp. 46-53, 2016.

[9] J. E. C. Cruz, E. H. Shiguemon, L. N. F. Guimarães, "A comparison of Haar-like, LBP and HOG approaches to concrete and asphalt runway detection in high resolution imagery", in *JCIS*, Dec, 20, 2015.

[10] Burcu Kır Savaş; Sümeyya İlkin; Yaşar Becerikli, "The realization of face detection and fullness detection in medium by using Haar Cascade

Classifiers", in *24th Signal Processing and Communication Application Conference (SIU)*. May,16-19,2019.

[11] Li Cuimei; Qi Zhiliang; Jia Nan; Wu Jianhua, "Human face detection algorithm via Haar cascade classifier combined with three additional classifiers", in *13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*. Oct,20-22,2017

[12] S. Yang, P. Luo, C. C. Loy, and X. Tang, "From facial parts responses to face detection: A deep learning approach," in *IEEE International Conference on Computer Vision*, pp. 3676–3684, 2015.

[13] T.-Y. Lin, P. Dollar, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.936–944, 2016.

[14] François Chollet. Xception, "Deep Learning with Depthwise Separable Convolutions", in *Computer Vision and Pattern Recognition 2016*.

[15] O. Arriaga, Paul G. Plöger, M. Valdenegro, "Real-time Convolutional Neural Networks for Emotion and Gender Classification," in *ICRA 2018*, Oct, 20, 2017.

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Computer Vision – ECCV 2016, Springer International Publishing, 2016*, pp. 21–37.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Computer Vision and Pattern Recognition*, 06 2016, pp. 779–788.

[18] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition 2014*, pp. 580–587, 2014.

[19] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5525–5533, 2016.

[20] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *in IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, Oct 2016.

[21] 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), pp. 815–823, 2015.

[22] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778, 2016

[23] Fares Jalled, “Face Recognition Machine Vision System Using Eigenfaces”, in *Computer Vision and Pattern Recognition, 2017*.

[24] Patrik K., Miroslav B., Tomas M., Roman R., “A New Method for Face Recognition Using Convolutional Neural Network”, in Digital image processing and computer graphics, vol. 15, pp. 663-672 (2017).

[25] X. Sun, P. Wu and Steven C.H. Hoi, “Face Detection using Deep Learning: An Improved Faster RCNN Approach”, in *2018 Neuro computing Journal*, Volume 299, pp. 42-50, 2018.

[26] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang and S. Z. Li, "FaceBoxes: A CPU real-time face detector with high accuracy," 2017 IEEE International Joint Conference on Biometrics (IJCB), Denver, CO, 2017, pp. 1-9, doi: 10.1109/BTAS.2017.8272675.

[27] E. Pedregosa, F. Varoquaux, G. Gramfort, A. Michel, V. Thirion, B. Grisel, O. Blondel, M. Prettenhofer, P. Weiss, R. Dubourg, V. Vanderplas, J. Passos, A. Cournapeau, D. Brucher, M. Perrot, M. Duchesnay: Scikit-learn: Machine Learning in Python. In: Journal of Machine Learning Research, vol 12, pp. 2825-2830, 2011.

[28] Wu. Xiang, Wu. Ning, “Thresholded Two-Phase Test Sample Representation for Outlier Rejection in Biological Recognition”, in *Computational and mathematical methods in medicine, Volume 2013*, Mar, 11, 2013.

[29] Logistic Regression. "From Wikipedia."

[30] Zhang Z., SongY., and Qi H.: Age Progression/Regression by Conditional Adversarial Autoencoder. In: CVPR 2017, pp. 4352-4360 (2017).

[31] Cambridge P. J. Phillips, H. Wechsler, and P. Rauss. The FERET database and evaluation procedure for face-recognition algorithms. Image and Vision Computing, 16(5):295–306, 1998.

[32] Georghiades, Athinodoros & Belhumeur, Peter & Kriegman, David. (2001). From Few to Many: Illumination Cone Models for Face Recognition Under Variable Lighting and Pose. IEEE Trans. Pattern Anal. Mach. Intell. 23. 643-

660. 10.1109/34.92746.

[33] AT&T Laboratories, The Database of Faces, April 1994.

Appendix A

❖ *Parameters of devices in the system*

1. Camera HIKVISION DS-2DE2A404IW-DE3 4MP Outdoor PTZ

Specification:

This PTZ camera is housed in an IP66- and IK10-rated

+ Resolution: 4MP
+ 1/3″ progressive scan CMOS
+ Up to 2560 × 1440@25fps resolution
+ Min. illumination:
  Color: 0.005 Lux @(F1.6, AGC ON)
  B/W: 0.001 Lux @(F1.6, AGC ON)
+ 4× optical zoom, 16× digital zoom
+ 120 dB WDR, 3D DNR, HLC, EIS, Regional Exposure, Regional Focus
+ 12 VDC & PoE
+ IK10, IP66
+ Support H.265+/H.265 video compression
+ Support Pan & Tilt
+ Many manual functions: White Balance, AGC, ...

2. AC1200 Smart Dual-Band WiFi Router

Specification:

Supported frequency bands: 2.4 GHz / 5 GHz

Connection standard: 802.11 a /b/g/n/ac

Compliant with IEEE802.11ac speed up to 1167Mbps (5GHz: 867Mbps + 2.4GHz: 300Mbps)

Antenna: 4x outside

Smart WiFi Schedule for lower power consumption

3. PC Server

Specification:

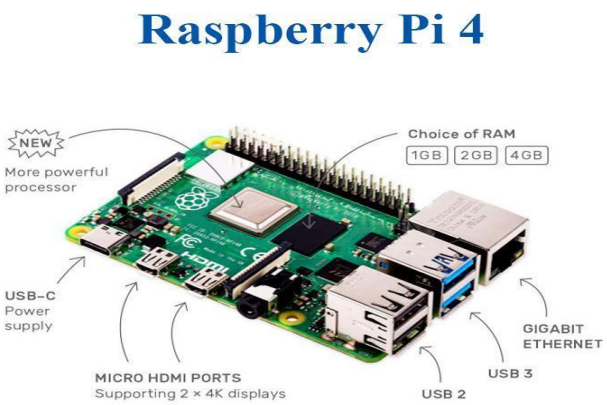
+ CPU Intel Core I5 9400F (2.9-4.1Ghz)
+ Mainboard Gigabyte B360M

+ Ram DDR4 16 GB-bus 3000Mhz
+ VGA Gigabyte RTX 2060 6Gb
+ SSD 240 GB
+ HDD 1 TB

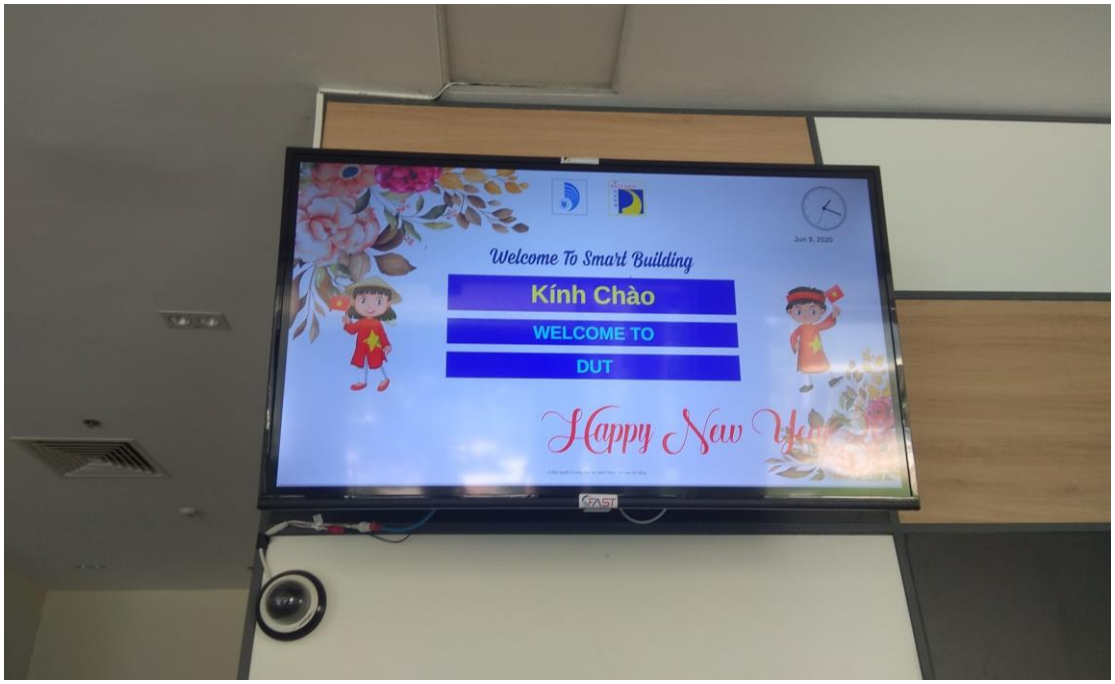
4. Tivi Led 50 inches

5. Raspberry pi 4

Using for access our web app



❖ ***Setup the system***



Appendix B

A. Table 1 and Table 2 show a detailed description of the number of photos of each individual in the self-built DUT dataset:

ID	Train	WM	HM	Sum	ID	Train	WM	HM	Sum
10087	24	10	209	243	11100	7	3	92	102
10125	168	72	811	1051	11102	41	18	261	320
10171	10	4	83	97	11109	8	3	14	25
10220	10	4	17	31	11118	20	8	538	566
10222	17	7	289	313	11160	14	6	68	88
10225	18	8	134	160	11184	7	3	45	55
10229	14	6	36	56	11185	7	3	28	38
10234	18	8	37	63	11193	15	6	485	506
10250	7	3	74	84	11201	21	9	162	192
10272	7	3	167	177	11214	13	5	45	63
10276	7	3	22	32	11216	22	9	147	178
10283	10	4	34	48	11226	34	15	442	491
10307	29	13	125	167	11243	13	5	234	252
10329	27	12	267	306	11245	13	5	170	188
10349	27	11	118	156	11248	7	3	52	62
10354	8	3	17	28	11249	25	10	161	196
10358	27	11	187	225	11251	37	16	163	216
10359	20	9	174	203	11267	7	3	51	61
10398	52	22	445	519	11272	10	4	210	224
10402	30	13	140	183	11273	13	6	117	136
10423	18	7	32	57	11280	7	3	68	78
10437	7	3	53	63	11281	43	18	153	214
10473	18	7	180	205	11317	21	9	72	102
10533	8	4	91	103	11318	19	8	77	104
10542	17	7	201	225	11321	7	3	34	44
10576	7	3	11	21	11409	10	4	72	86
10600	8	4	107	119	11410	7	3	10	20
10609	8	4	81	93	11455	7	3	46	56
10635	7	3	28	38	11456	44	19	328	391
10766	31	13	229	273	11458	27	11	219	257
10824	15	7	55	77	11468	10	4	43	57
10861	7	3	7	17	11489	37	16	57	110
10872	15	6	112	133	11499	17	7	54	78
10887	10	4	164	178	11500	47	20	140	207
10888	27	11	77	115	193019	28	12	40	80
10966	15	6	91	112	1149822	7	3	25	35
11070	16	7	45	68	1641764600	17	7	161	185
11072	7	3	110	120	8493773300	19	8	261	288
11074	7	3	37	47	9563444161	24	10	168	202
11088	29	13	202	244	9944005072	22	9	110	141

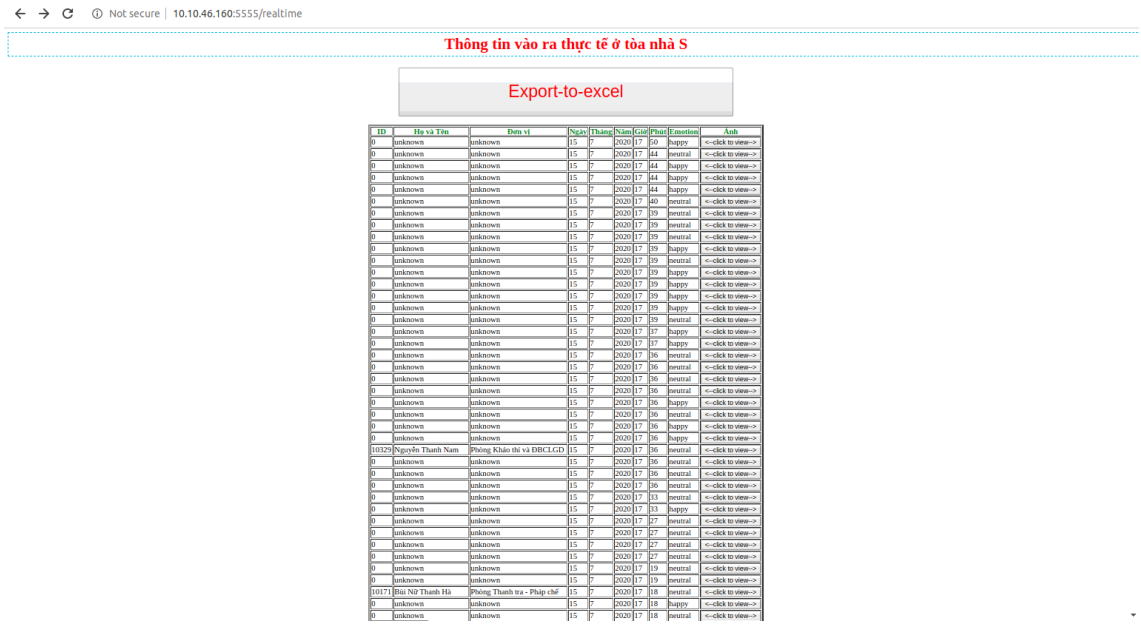
Table 1 Describe the number of images for each individual

ID	Center	Left	Right	Down	ID	Center	Left	Right	Down
10087	141	0	11	91	11100	56	0	11	35
10125	842	0	51	158	11102	197	0	61	62
10171	52	0	31	14	11109	10	2	13	0
10220	16	0	0	15	11118	318	21	169	58
10222	107	32	99	75	11160	62	0	0	26
10225	66	0	39	55	11184	23	2	20	10
10229	42	0	14	0	11185	16	0	7	15
10234	19	0	33	11	11193	149	120	48	189
10250	52	4	8	20	11201	112	1	4	75
10272	79	11	45	42	11214	14	0	28	21
10276	1	0	12	19	11216	173	0	18	187
10283	43	0	2	3	11226	246	2	39	204
10307	69	0	14	84	11243	54	35	33	130
10329	236	4	42	26	11245	77	10	3	98
10349	89	20	33	14	11248	10	52	0	0
10354	9	0	7	12	11249	77	30	30	59
10358	130	8	15	74	11251	43	0	0	173
10359	95	0	2	106	11267	15	0	46	0
10398	352	6	73	88	11272	98	0	29	97
10402	113	1	55	13	11273	39	0	0	97
10423	18	0	8	31	11280	73	0	0	5
10437	38	0	0	25	11281	100	0	17	97
10473	115	17	73	0	11317	79	3	6	14
10533	22	16	0	65	11318	41	0	60	3
10542	131	2	56	36	11321	27	0	1	16
10576	8	4	3	6	11409	20	9	31	26
10600	34	9	22	54	11410	2	0	18	0
10609	52	26	12	3	11455	14	0	8	34
10635	30	0	8	0	11456	192	0	50	149
10766	60	0	37	176	11458	115	0	59	83
10824	20	0	15	42	11468	34	0	0	23
10861	12	0	5	0	11489	85	0	21	4
10872	39	0	84	10	11499	54	0	24	0
10887	108	0	6	64	11500	113	0	61	33
10888	57	0	58	0	193019	21	8	44	7
10966	40	0	0	72	1149822	22	0	13	0
11070	10	0	58	0	1641764600	127	0	45	13
11072	48	11	36	24	8493773300	228	0	51	9
11074	16	0	0	31	9563444161	134	0	49	19
11088	136	11	2	95	9944005072	97	0	12	32

Table 2 Description of the developed database

B. Details on options for searching for information on people entering the building.

*Information in “Real-time”



ID	Họ và Tên	Đơn vị	Ngày	Tháng	Năm	Giờ	Phút	Emotion	Ảnh
0	unknown	unknown	15	7	2020	17	50	happy	<--click to view-->
0	unknown	unknown	15	7	2020	17	44	neutral	<--click to view-->
0	unknown	unknown	15	7	2020	17	44	happy	<--click to view-->
0	unknown	unknown	15	7	2020	17	44	happy	<--click to view-->
0	unknown	unknown	15	7	2020	17	44	happy	<--click to view-->
0	unknown	unknown	15	7	2020	17	40	neutral	<--click to view-->
0	unknown	unknown	15	7	2020	17	39	neutral	<--click to view-->
0	unknown	unknown	15	7	2020	17	39	neutral	<--click to view-->
0	unknown	unknown	15	7	2020	17	39	neutral	<--click to view-->
0	unknown	unknown	15	7	2020	17	39	happy	<--click to view-->
0	unknown	unknown	15	7	2020	17	39	neutral	<--click to view-->
0	unknown	unknown	15	7	2020	17	39	happy	<--click to view-->
0	unknown	unknown	15	7	2020	17	39	happy	<--click to view-->
0	unknown	unknown	15	7	2020	17	39	happy	<--click to view-->
0	unknown	unknown	15	7	2020	17	39	neutral	<--click to view-->
0	unknown	unknown	15	7	2020	17	37	happy	<--click to view-->
0	unknown	unknown	15	7	2020	17	37	happy	<--click to view-->
0	unknown	unknown	15	7	2020	17	36	neutral	<--click to view-->
0	unknown	unknown	15	7	2020	17	36	neutral	<--click to view-->
0	unknown	unknown	15	7	2020	17	36	neutral	<--click to view-->
0	unknown	unknown	15	7	2020	17	36	neutral	<--click to view-->
0	unknown	unknown	15	7	2020	17	36	happy	<--click to view-->
0	unknown	unknown	15	7	2020	17	36	neutral	<--click to view-->
0	unknown	unknown	15	7	2020	17	36	happy	<--click to view-->
0	unknown	unknown	15	7	2020	17	36	happy	<--click to view-->
10129	Nguyễn Thanh Nam	Phòng Khảo thí và ĐBCLGD	15	7	2020	17	36	neutral	<--click to view-->
0	unknown	unknown	15	7	2020	17	36	neutral	<--click to view-->
0	unknown	unknown	15	7	2020	17	36	neutral	<--click to view-->
0	unknown	unknown	15	7	2020	17	36	neutral	<--click to view-->
0	unknown	unknown	15	7	2020	17	33	neutral	<--click to view-->
0	unknown	unknown	15	7	2020	17	33	happy	<--click to view-->
0	unknown	unknown	15	7	2020	17	27	neutral	<--click to view-->
0	unknown	unknown	15	7	2020	17	27	neutral	<--click to view-->
0	unknown	unknown	15	7	2020	17	27	neutral	<--click to view-->
0	unknown	unknown	15	7	2020	17	27	neutral	<--click to view-->
0	unknown	unknown	15	7	2020	17	19	neutral	<--click to view-->
0	unknown	unknown	15	7	2020	17	19	neutral	<--click to view-->
10171	Bùi Nữ Thanh Hà	Phòng Thanh tra - Pháp chế	15	7	2020	17	18	neutral	<--click to view-->
0	unknown	unknown	15	7	2020	17	18	happy	<--click to view-->
0	unknown	unknown	15	7	2020	17	18	neutral	<--click to view-->

Figure 1. Real-time View.

*See according to Unit/Person of Unit:

ID	Họ Và Tên	Đơn Vị	Ngày Sinh	Giới Tính	Chức Vụ	Trình độ	Học hàm	Ngày	Tháng	Năm	Giờ	Ảnh record
9944005072	Nguyễn Lương Quảng	Phòng Nghiên Cứu DNIIT		Nam	Sinh viên	Đại học		13	1	2020	15	<--click to view-->
8493773300	Lã Tấn Đạt	Phòng Nghiên Cứu DNIIT	11/07/1997	Nam	Sinh Viên	Đại học		13	1	2020	16	<--click to view-->
9944005072	Nguyễn Lương Quảng	Phòng Nghiên Cứu DNIIT		Nam	Sinh viên	Đại học		13	1	2020	16	<--click to view-->
1641764600	Trần Quang Huy	Phòng Nghiên Cứu DNIIT		Nam	Sinh viên	Đại học		13	1	2020	16	<--click to view-->
9563444161	Lê Tiến Nhật	Phòng Nghiên Cứu DNIIT	28/10/1997	Nam	Sinh viên	Đại học		13	1	2020	16	<--click to view-->
1641764600	Trần Quang Huy	Phòng Nghiên Cứu DNIIT		Nam	Sinh viên	Đại học		13	1	2020	17	<--click to view-->
8493773300	Lã Tấn Đạt	Phòng Nghiên Cứu DNIIT	11/07/1997	Nam	Sinh Viên	Đại học		13	1	2020	17	<--click to view-->
9563444161	Lê Tiến Nhật	Phòng Nghiên Cứu DNIIT	28/10/1997	Nam	Sinh viên	Đại học		13	1	2020	17	<--click to view-->
9563444161	Lê Tiến Nhật	Phòng Nghiên Cứu DNIIT	28/10/1997	Nam	Sinh viên	Đại học		13	1	2020	18	<--click to view-->
9563444161	Lê Tiến Nhật	Phòng Nghiên Cứu DNIIT	28/10/1997	Nam	Sinh viên	Đại học		14	1	2020	10	<--click to view-->
9944005072	Nguyễn Lương Quảng	Phòng Nghiên Cứu DNIIT		Nam	Sinh viên	Đại học		14	1	2020	10	<--click to view-->
8493773300	Lã Tấn Đạt	Phòng Nghiên Cứu DNIIT	11/07/1997	Nam	Sinh Viên	Đại học		14	1	2020	10	<--click to view-->
8493773300	Lã Tấn Đạt	Phòng Nghiên Cứu DNIIT	11/07/1997	Nam	Sinh Viên	Đại học		14	1	2020	11	<--click to view-->
1641764600	Trần Quang Huy	Phòng Nghiên Cứu DNIIT		Nam	Sinh viên	Đại học		14	1	2020	11	<--click to view-->
9944005072	Nguyễn Lương Quảng	Phòng Nghiên Cứu DNIIT		Nam	Sinh viên	Đại học		14	1	2020	11	<--click to view-->
9563444161	Lê Tiến Nhật	Phòng Nghiên Cứu DNIIT	28/10/1997	Nam	Sinh viên	Đại học		14	1	2020	11	<--click to view-->
9563444161	Lê Tiến Nhật	Phòng Nghiên Cứu DNIIT	28/10/1997	Nam	Sinh viên	Đại học		14	1	2020	12	<--click to view-->
8493773300	Lã Tấn Đạt	Phòng Nghiên Cứu DNIIT	11/07/1997	Nam	Sinh Viên	Đại học		14	1	2020	12	<--click to view-->
9944005072	Nguyễn Lương Quảng	Phòng Nghiên Cứu DNIIT		Nam	Sinh viên	Đại học		14	1	2020	12	<--click to view-->

Figure 2. View according to Unit/Person

*Details information of Person/Unit:

Figure 3. The detailed information display.

*Learning new personal: There are 3 option building front-in Web App

a. From camera (recording video)
b. Upload a video (Format .avi or .mp4)
c. Upload Multi-Images.

Figure 4. Information required when learning new people.

Figure 5. Options for learning new people.

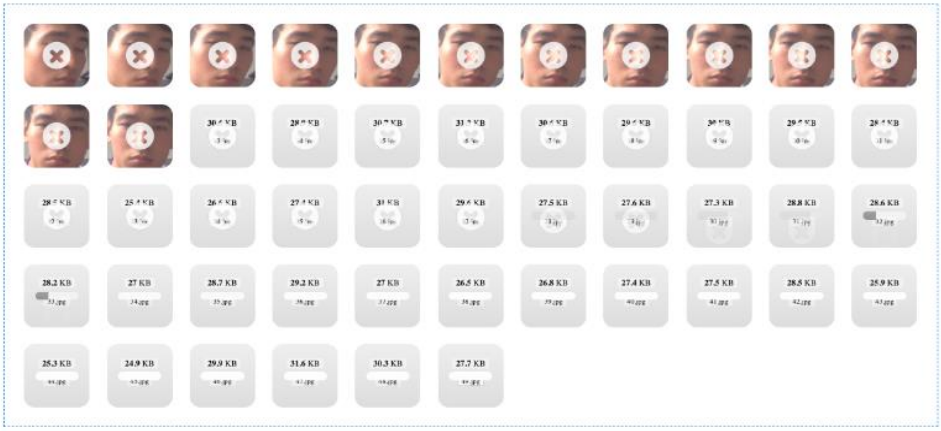


Figure 6. Processor upload to images.

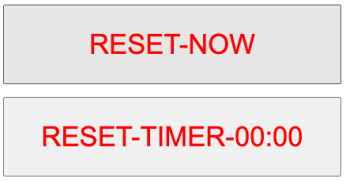


Figure 7. Reset the system when new people are finished learning.