Ministry of Higher Education

Misr Higher Institute For Commerce & Computer

Computer Science Department

Graduation project

Smart Student Attendance System

2021

By

Abdelrhman Talat Abdelrhman

Aishaa Gamal Ahmed Ali

Ammar Yasser Werdani

Mohamed Ayman Hamed

Omar Elsayed Abdlnasser

Omar Shokri Al Rajhi

Supervisor

Assoc.Prof.DR.Noha Heikal

Assistant

Eng.Nehad Elsharkawy

ACKNOWLEDGMENTS

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project.

All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank **Assoc.Prof.DR.Noha Heikal** for providing us an opportunity to do the project work, and giving us all support and guidance which made us complete the project duly.

We are extremely thankful to her for providing such a nice support and guidance.

We thank **Eng. Nehad El Sharkawy**, who provided us with some advice and guidance until the completion of our project by providing some information to develop a good system.

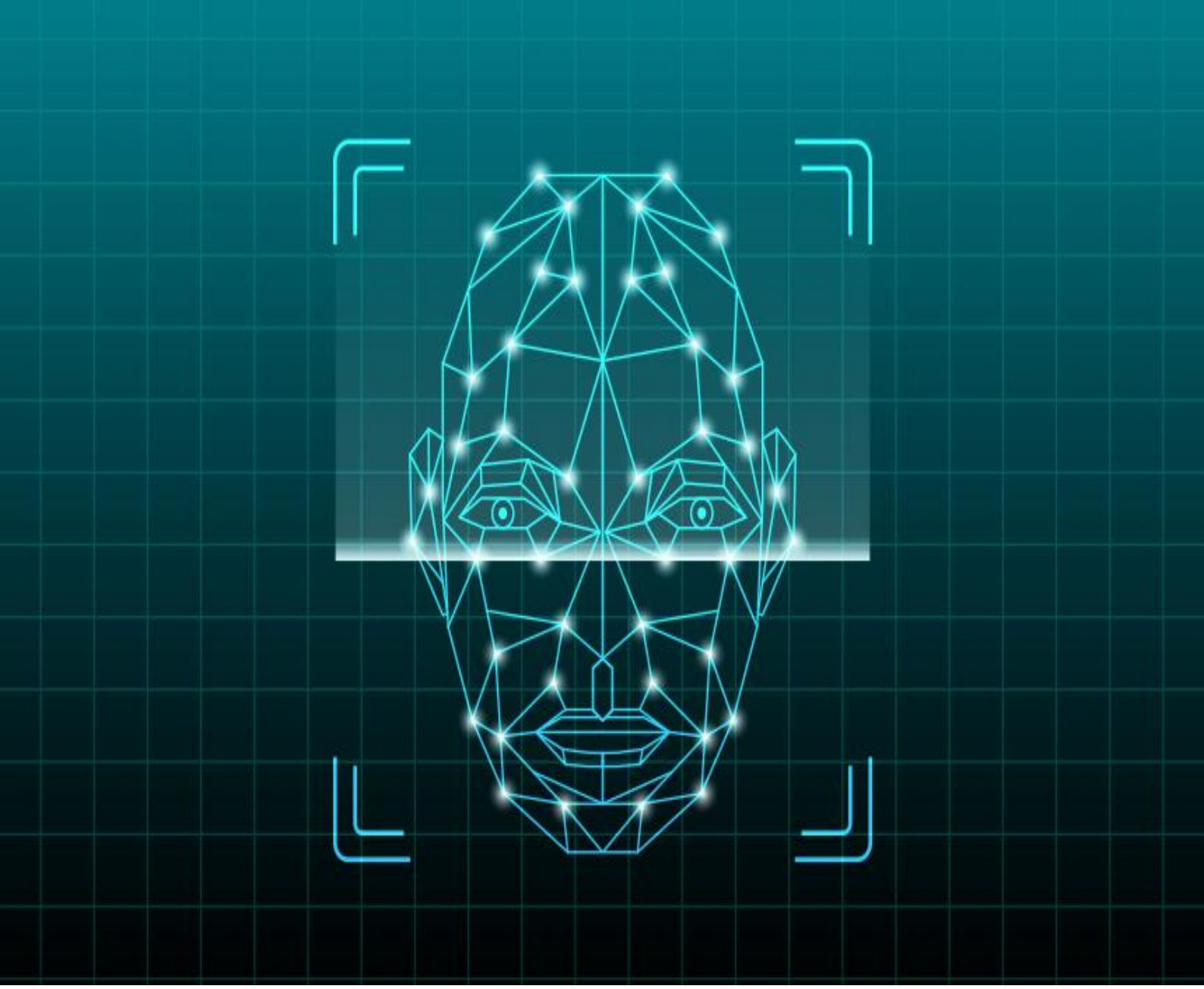
ABSTRACT

We wanted to create something that helps protect the students as well as the teacher in light of the current circumstances and the spread of the new [**Corona virus**], so we created the Smart Student Attendance System.

It is a facial recognition system that facilitates recording the daily attendance of the student by taking a picture of his face in an easy way to interact with it.

We also wanted to make sure that the system is as understandable and smooth as possible. In the end, we want to save time and effort, as well as reduce the possibility of infection as much as possible.

INDEX



CHAPTER 1

INTRODUCTION

Student attendance system

uses real-world images(screenshots, Internet images, or photographs) as the stimuli for searches, Modern visual search technology uses AI (artificial intelligence) to understand the content and context of these images and return alist of related results and, Face recognition, as one of the most successful applications of image analysis, has recently gained significant attention.

It is due to availability of feasible technologies, including mobile solutions. Research in automatic. Although systems have been developed for face detection and tracking, reliable face recognition still offers a great challenge to computer vision and pattern recognition researchers. There are several reasons for recent increased interest in face recognition, including rising public concern for security, the need for identity verification in the digital world, face analysis and modelling techniques in multimedia data management and computer entertainment.

Face recognition is a method of identifying or verifying the identity of an individual using their face. Face recognition systems can be used to identify people in photos, video, or in real-time.

In this days the use of facial recognition has become very common, whether for the work market, daily life, or for entertainment, as some sites use it to find your likeness of other nationalities or even to reunite twins who have been separated between them in the past in addition to recording attendance and absence in some facilities or Law enforcement may also use mobile devices to identify people during police stops.

1.1 Motivation & Overview

The motivation behind building this project is that it is one of the most successful.

It is important for universities because it is in light of the spread of the new Corona virus [Covid 19].

This project provides safety as it reduces the risks students are exposed to becouse :

- It saves time.
- Reduces congestion.
- It also prevents any error in recording students' attendance.
- Automatic photo capture.

1.2 Main Problem

Security at the university

Visual research is important for universities because it provides safety and security, and reduces risk and congestion.

It also answers the question, "Is this individual a member of the organization or not?"

Examines the member's face and confirms whether or not he is actually a member of that organization.

If he is registered with the institution’s data, then he is allowed to enter, and if he is not registered with the institution, he is not allowed to enter

We can use this feature in the event that the member does not own or lose the institution's membership card, in which case face recognition is used.

1.3 Objectives

The visual search or facial recognition

- **provides** easy logging in to the facilities, once you stand for a moment in front of the camera that records a live video clip and as soon as you get a suitable shot of your face, the program analyzes the image and extracts the distinctive features of your face and then compares it to the particles saved in the database associated with the system.

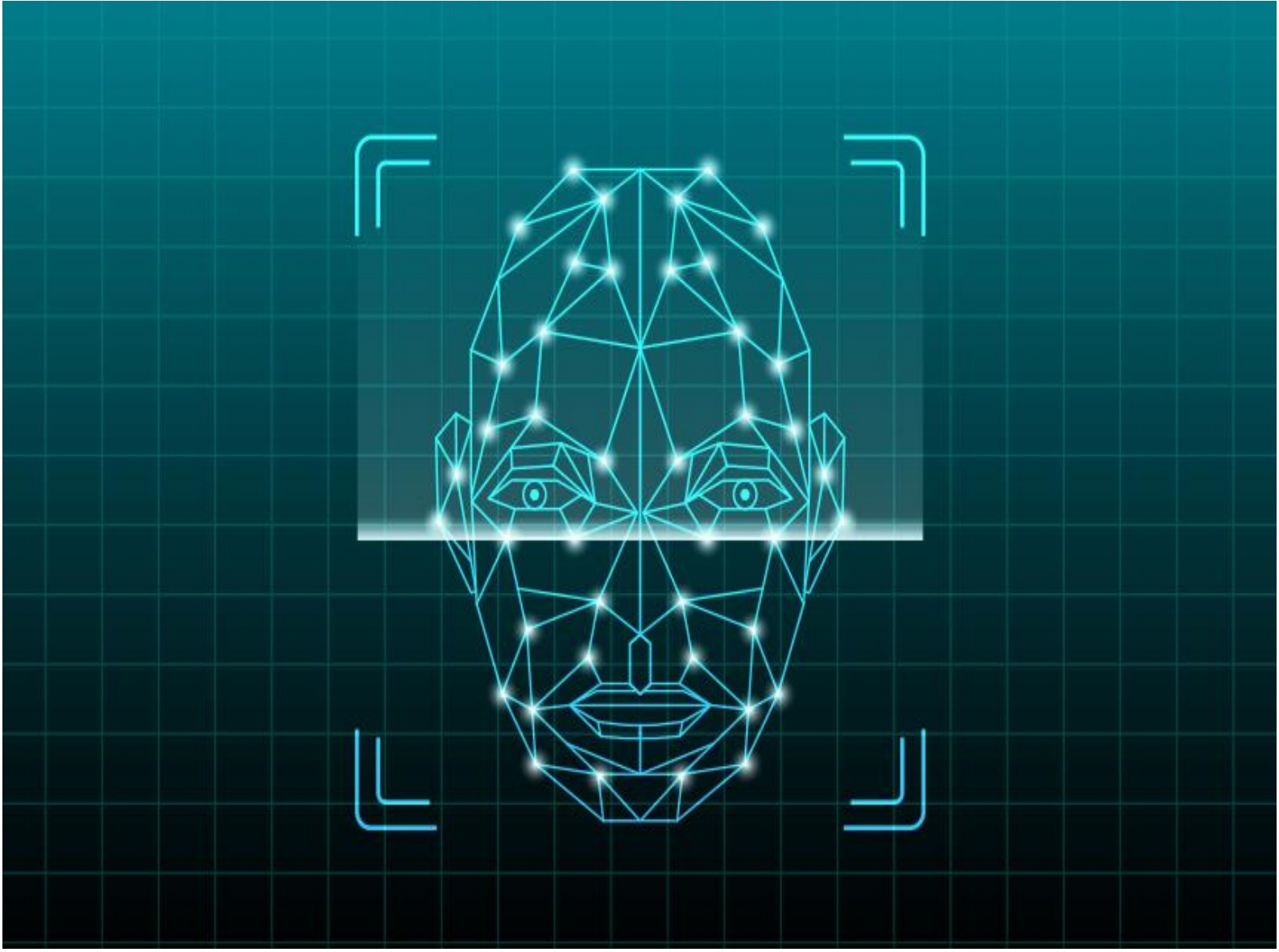
- **After** making sure that these features are present in the institution's database, the owner of these features logs into his statute, which will be the student, as well as on the statute of the responsible admin, who will be the doctor or instructor in this case.

1.4 Tools and Methods

- Python.
- OpenCV.
- Face recognition.
- pyQt5.

1.5 Project Outline

- Chapter 2 will explain a background of the tools used to develop the system.
- Chapter 3 will explain the analysis of the system , it include a full analysis of the system , diagrams of the system.
- Chapter 4 will introduce the steps to implement the system.
- Chapter 5 will introduce the conclusion and future work of this project.



CHAPTER 2

BACKGROUND

2.1 Introduction

In this chapter we will know what is the Visual search,difference between Visual search and Image search,some of the applications based on it, Face recognation,Deep learning, what is DNN,CNN,Real time face recogntion, Face detection,openCV, PyQt5.

2.2 what is visual search?

Visual search is a goal oriented activity that occurs regularly in daily life and involves the active scanning of the environment in order to locate a particular target among irrelevant non-targets, or distractors.

Visual search uses artificial intelligence technology to help people search through the use of real-world imagery, rather than through text search.

So, when a person takes a photograph of an object, using Google Lens, for instance, the software identifies the object within the picture and provides information and search results to the user.

This technology is particularly useful for eCommerce stores and brands, and with the implementation of well-optimized content, they could stand the chance of being the returned search result for a user.[14]

2.3 Why Visual Search Differs From Image Search?

Visual search falls under the umbrella of what is known as "sensory search", which includes searching via text, voice, and vision.

Although both visual and image search is based around imagery, the crucial difference lies in the fact people use words to conduct an image search, whereas, with visual search, a user uses an image to conduct the search.

Visual search works through a combination of computer vision and machine learning technology.

Computer vision not only allows the machines to see, but it helps them to interpret what they are seeing, before coming to a decision about what to do with the information.

As a technology, computer vision has been around for a long-time, but it wasn't until the advancement of machine learning technology that it could be used for visual search.

In essence, machine learning provides information that computer vision needs to understand what is presented within an image.

With Google Lens, the technology cross-references information with what is known within the Google Knowledge Graph. Other platforms use other software to support their own versions of image recognition technology.[6]

2.4 Some application based on visual search

2.4.1 Pinterest Lens

Introduced in 2017, Pinterest Lens boasts over 600 million searches every month, and it was announced in 2019 that the technology has the ability to recognize over 2.5 billion items.

The USP of Pinterest's visual search app is that users can take a photo of almost anything, and they will be able to find, save, or shop for the items contained within the picture.

A powerful ally of fashion brands, retailers, and home decor companies alike, Pinterest states that 90% of users' purchasing decisions are informed by the information attained through the technology.

Figure 1 Pinterest

2.4.2 Bing Visual Search

Bing Visual Search technology is a very different visual search tool from Pinterest Lens, due to the fact that it works to provide people with information as well as products; pretty much like the Bing search engine itself.

That said, when shopping intent is detected, Bing Visual Search also provides users with associated products and pricing information.

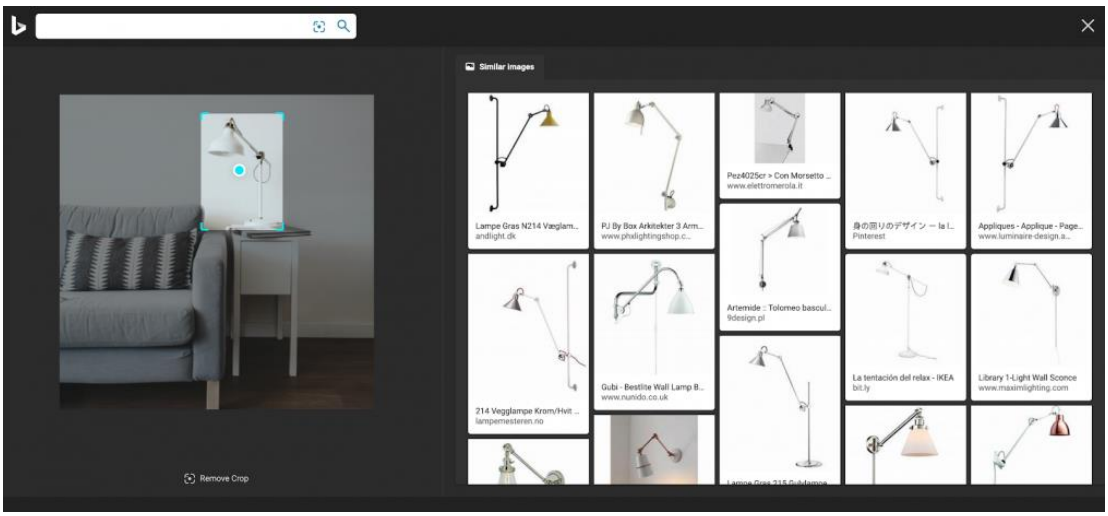


Figure 2 Bing Visual Search

2.4.3 Snapchat Camera Search

In September 2018, Snapchat announced the arrival of Snapchat Camera Search, which invited users to search for products on Amazon using the app.

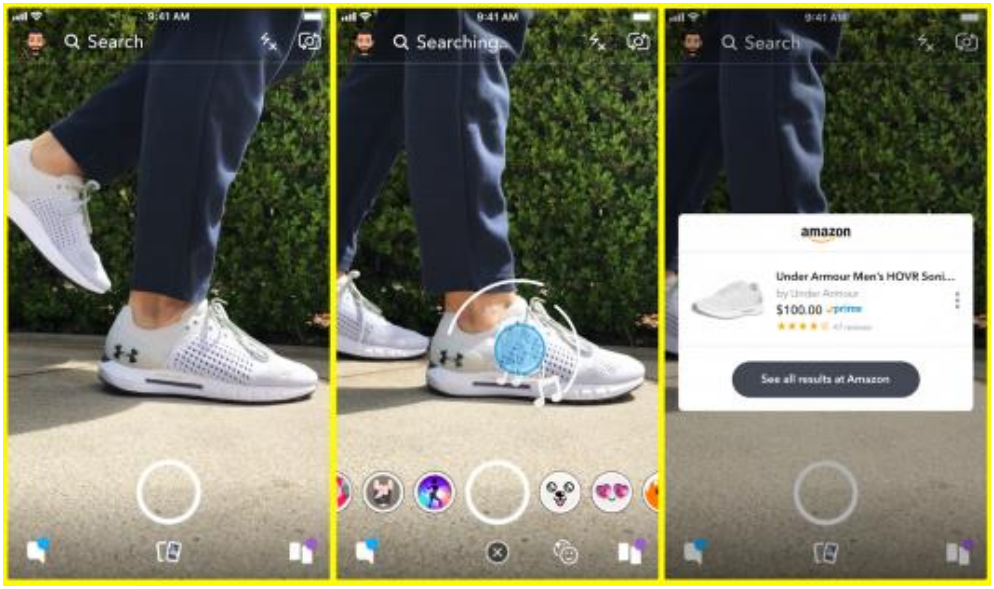


Figure 2 Snapchat camera search

When the app recognizes a barcode, an Amazon card is presented to the user, offering them a link for that product or a selection of similar items from the Amazon store. If a user has the Amazon app installed, they can tap the app and directed straight to the page within the Amazon app.

2.4.4 Google Lens

Last but certainly not least, Google Lens was announced and launched at Google I/O in 2017 and has quickly become the most popular visual search platform in the world because it features advanced visual search capabilities.

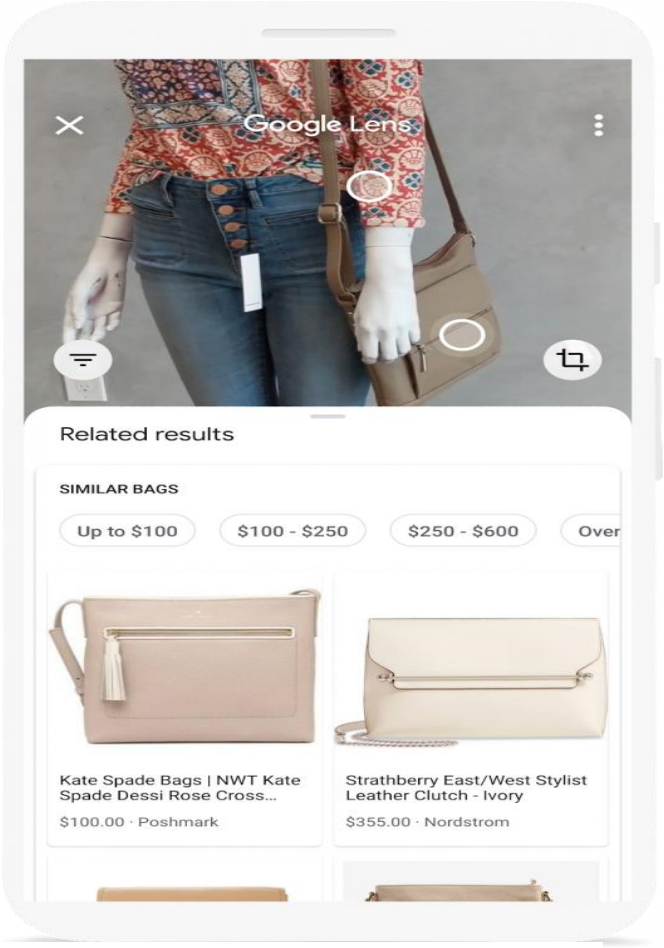


Figure 3 Google lens

2.5 What is Face Recogntion?

It is a technology that is able to automatically identify or verify a person in a digital photo or video frame taken from a video clip. There are many ways that facial recognition systems work, but in general they compare the facial features selected in a particular photo with the faces stored in a database.

Face recognition is a broad problem of identifying or verifying people in photographs and videos. Face recognition is a process comprised of detection, alignment, feature extraction, and a recognition task Deep learning models first approached then exceeded human performance for face recognition tasks.[13]

2.5.1 Faces in Photographs

There is often a need to automatically recognize the people in a photograph. There are many reasons why we might want to automatically recognize a person in a photograph. For example: We may want to restrict access to a resource to one person, called face authentication. We may want to confirm that the person matches their ID, called face verification. We may want to assign a name to a face, called face identification. Generally,

we refer to this as the problem of automatic “face recognition” and it may apply to both still photographs or faces in streams of video. Humans can perform this task very easily. We can find the faces in an image and comment as to who the people are, if they are known. We can do this very well, such as when the people have aged, are wearing sunglasses, have different colored hair, are looking in different directions, and so on. We can do this so well that we find faces where there aren’t any, such as in clouds.[18]

2.5.2 Some application based on face recogntion

1- F1 Face Recognition Attendance and Access Control Terminal

F1 Is a Convenient And Safe Industrial-Grade Visual Face Recognition Terminal Based on the Latest Face Recognition Algorithm of Baidu.

F1 Face Recognition Verification Terminal Can Realize Personnel Identity Verification, Personnel Entry and Exit Management and Attendance Management.

The System Supports Binocular Infrared Live Body Detection with the Function of Temperature Measuring.

2- Availo App

It is a system for adjusting employee working hours using Internet of Things technology through the face or voice fingerprint of each employee.

You do not need more than a smartphone to ensure employee attendance and departure on time

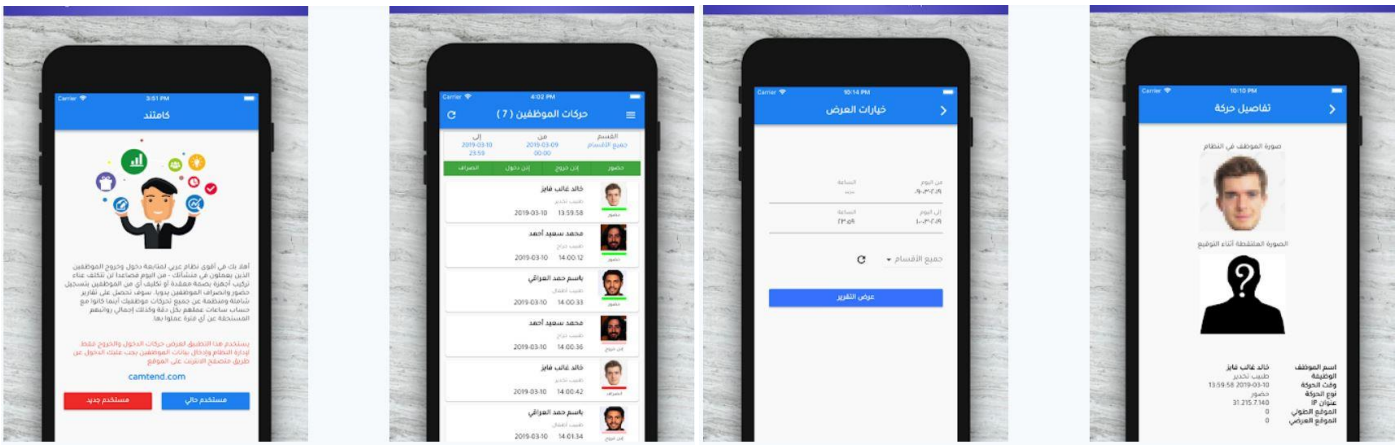
Availo, is an attendance app that replaces the traditional fingerprint machines.

The app helps employers monitor their employees' working hours and help them achieve the daily required hours, especially working from home through the covid-19 pandemic.

3- camtend

It is a powerful and modern electronic system used to record attendance, leave movements and employee permissions on a central server through the employee's mobile phone or through a tablet or even a regular computer placed at the entrance to the company where a picture of the employee is taken and his location is determined when his fingerprint is taken.

Management can view, track and manage daily attendance data from anywhere at any time.



2.5.2 Process of Automatic Face Recognition

Face recognition is often described as a process that first involves four steps; they are: face detection, face alignment, feature extraction, and finally face recognition. Face Detection. Locate one or more faces in the image and mark with a bounding box. Face Alignment. Normalize the face to be consistent with the database, such as geometry and photometrics. Feature Extraction. Extract features from the face that can be used for the recognition task. Face Recognition. Perform matching of the face against one or more known faces in a prepared database. A given system may have a separate module or program for each step, which was traditionally the case, or may combine some or all of the steps into a single process.[18]

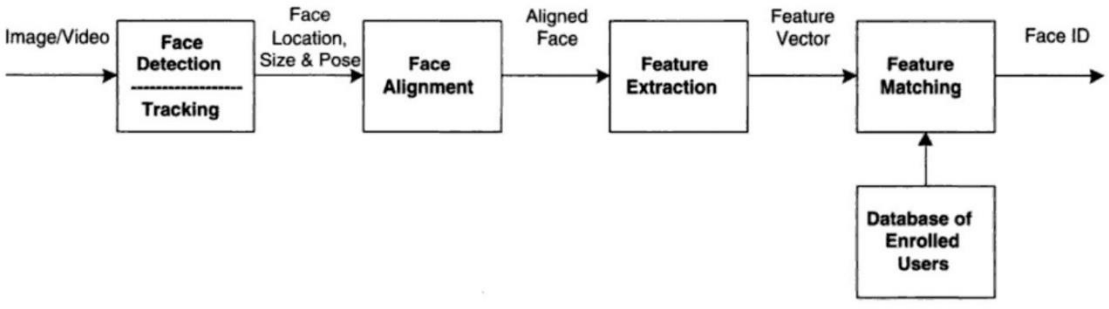


Figure 4 the Steps in a Face Recognition Process. Taken from "Handbook of Face Recognition," 2011.

2.5.3 Face Detection Task

Face detection is the non-trivial first step in face recognition. It is a problem of object recognition that requires that both the location of each face in a photograph is identified (e.g. the position) and the extent of the face is localized (e.g. with a bounding box). Object recognition itself is a challenging problem, although in this case, it is similar as there is only one type of object, e.g. faces, to be localized, although faces can vary wildly. The human face is a dynamic object and has a high degree of variability in its appearance, which makes face detection a difficult problem in computer vision. Further, because it is the first step in a broader face recognition system, face detection must be robust. For example, a face cannot be recognized if it cannot first be detected. That means faces must be detected with all manner of orientations, angles, light levels, hairstyles, hats, glasses, facial hair, makeup, ages, and so on as a visual

front-end processor, a face detection system should also be able to achieve the task regardless of illumination, orientation, and camera distance.[18]

2.5.4 Face Recognition Tasks

The task of face recognition is broad and can be tailored to the specific needs of a prediction problem. there are three face recognition tasks:

- ✓ **Face Matching:** Find the best match for a given face.
- ✓ **Face Similarity:**Find faces that are most similar to a given face.
- ✓ **Face Transformation**: Generate new faces that are similar to a given face.

And we can summarize these three separate tasks as follows:

Matching requires that the candidate matching face image be in some set of face images selected by the system. Similarity detection requires in addition to matching that images of faces be found which are similar to a recalled face this requires that the similarity measure used by the recognition system closely match the similarity measures used by humans Transformation applications require that new images created by the system be

similar to human recollections of a face. we have two main modes for face recognition, as:

- **Face Verification**. A one-to-one mapping of a given face against a known identity (e.g. is this the person?).
- **Face Identification**. A one-to-many mapping for a given face against a database of known faces (e.g. who is this person?).

A face recognition system is expected to identify faces present in images and videos automatically. It can operate in either or both of two modes:

1) face verification (or authentication).

2) face identification (or recognition).

We can describe the problem of face recognition as a supervised predictive modeling task trained on samples with inputs and outputs.[18]

In all tasks, the input is a photo that contains at least one face, most likely a detected face that may also have been aligned.

The output varies based on the type of prediction required for the task

For example:

It may then be a binary class label or binary class probability in the case of a face verification task.

It may be a categorical class label or set of probabilities for a face identification task.

It may be a similarity metric in the case of a similarity type task.

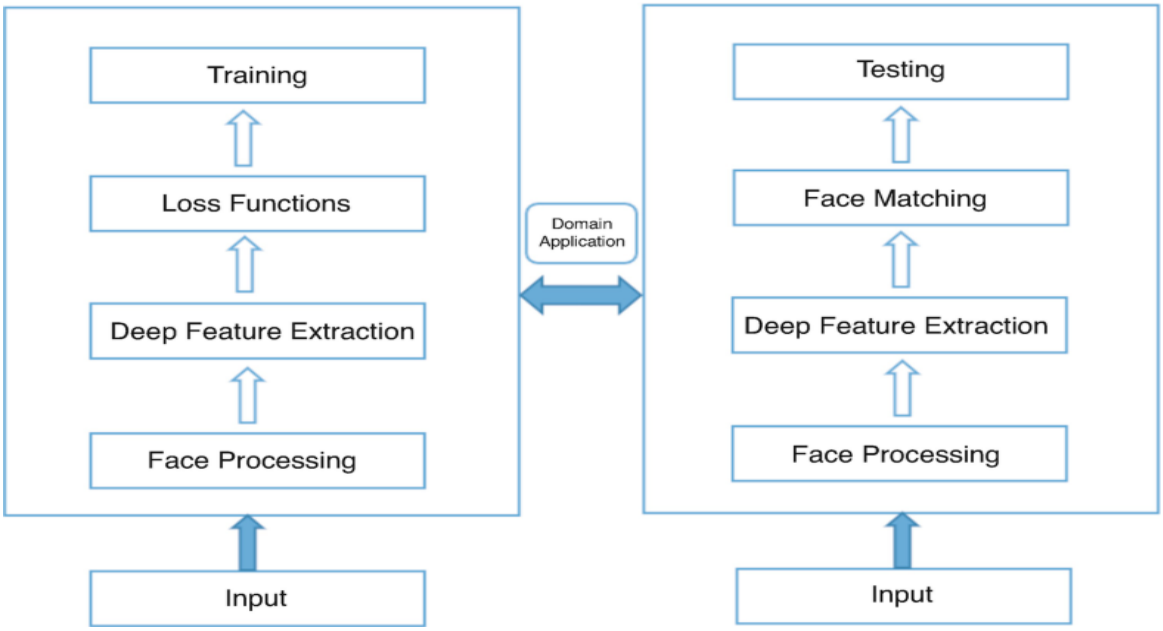


Figure 5 Face recognition with its different components with respect to training and testing phases

2.6 What is Deep learning?

Deep learning or deep structured learning is a new area of research on finding theories and algorithms for the familiarity machine to learn by itself by simulating the simulations of neurons in the human body and one of the branches of science mentioned in editing.

A branch of machine learning science, most diminishing learning stocks focus on falling copies of a high degree of contestants.

The discoveries in this area are proven with substantial volume support, fast and effective home page efficacy including facial recognition, speech recognition, computer vision, and natural language processing.[13]

2.6.1 Face Recognition with Deep Learning

In this section important deep architectures for face recognition

systems which have grown over the years are highlighted. The deep face recognition architectures are coupled with discriminative loss functions and face matching though deep features.

There are a large number of human faces available.

As such the face recognition process can be considered as fine grained object classification task.

However, for many applications it becomes really difficult to have all candidate faces during training stage.

This makes face recognition a zero shot learning activity.

As almost all the human faces share identical shape and texture, the representation adopted from small proportion of faces can generalize well.[2]

This calls towards including many IDs in the training set.

Various organizations have revealed that their deep face recognition system is trained by around .

The currently available public training databases consist

This issue is addressed through designing effective loss functions and deeper architectures in order to give deep features higher discrimination capability through small training datasets.

Here we place significant deep networks towards face recognition which has evolved over the years. The deep

networks considered here are DeepID, DeepID2, DeepID2+, DeepID3,DeepFace, Face++, FaceNet, and Baidu.

2.6.2 What is DNN platform?

DNN Platform, formerly called DotNetNuke Community Edition, is a free, open source content management system (CMS).

The DNN Platform is built on the .NET framework and is designed to be easy to use, without requiring extensive programming knowledge. Features of the DNN Platform include design skins that allow the appearance of a website

to be changed easily and the ability to incorporate third-party modules to add additional functionalities.

Because the CMS supports Multi-tenancy, multiple sites can be built on the same architecture.[2]

The DNN platform was originally based on IBuySpy Workshop, a program created by Shaun Walker as an extension to Microsoft's open source IBuySpy Portal. IBuySpy Portal, built on .NET Framework 1.0, was a reference application for building sites and applications using ASP.NET. In 2003, version 1.0.5 of iBuySpy Workshop was rebranded as DotNetNuke.

Other early versions of DotNetNuke included:

- DotNetNuke 2.0, released March 2004
- DotNetNuke 3.0, released March 2005
- DotNetNuke 4.0, released November 2005, which moved DotNetNuke from ASP.NET 1.1 to ASP.NET 2.0
- DotNetNuke 5.0, released December 2008

In September 2006, Walker co-founded DotNetNuke Corporation with Scott Willhite, Joe Brinkman and Nik Kalyani. In 2009, DotNetNuke Corporation launched DotNetNuke Professional Edition, a commercial version of the CMS. DotNetNuke Professional Edition included customer support provided by DotNetNuke Corporation, along with features not available in the open source software.

In July 2013, the company and the open source project were rebranded. DotNetNuke Corporation became DNN Corp. and DotNetNuke Community Edition took on its current name, DNN Platform. The DotNetNuke Professional Edition was rebranded and is now called Evoq Content. DNN Corporation continues to be the steward of the DNN Platform open source software project and sells the Evoq line of CMS products.[7]

2.6.3 What is Convolutional Neural Networks CNN?

Convolutional neural networks (ConvNets or CNNs) are a type of neural network that has been proven to be effective in various fields such as image recognition and classification.[8]

Filter neural networks have great success in distinguishing faces, inanimate objects, and traffic signals.[2]

a soccer player is kicking a soccer ball | a street sign on a pole in front of a building | a couple of giraffe standing next to each

Figure 6

In Figure.7. above, filtering neural networks are able to distinguish scenes and suggest appropriate titles for the image (example: a soccer player hitting a soccer ball), while in Figure (8), filtering

Neural networks can distinguish inanimate objects, animals, and even humans. In addition, these networks are effective in processing natural languages, such as analyzing or classifying linguistic phrases.

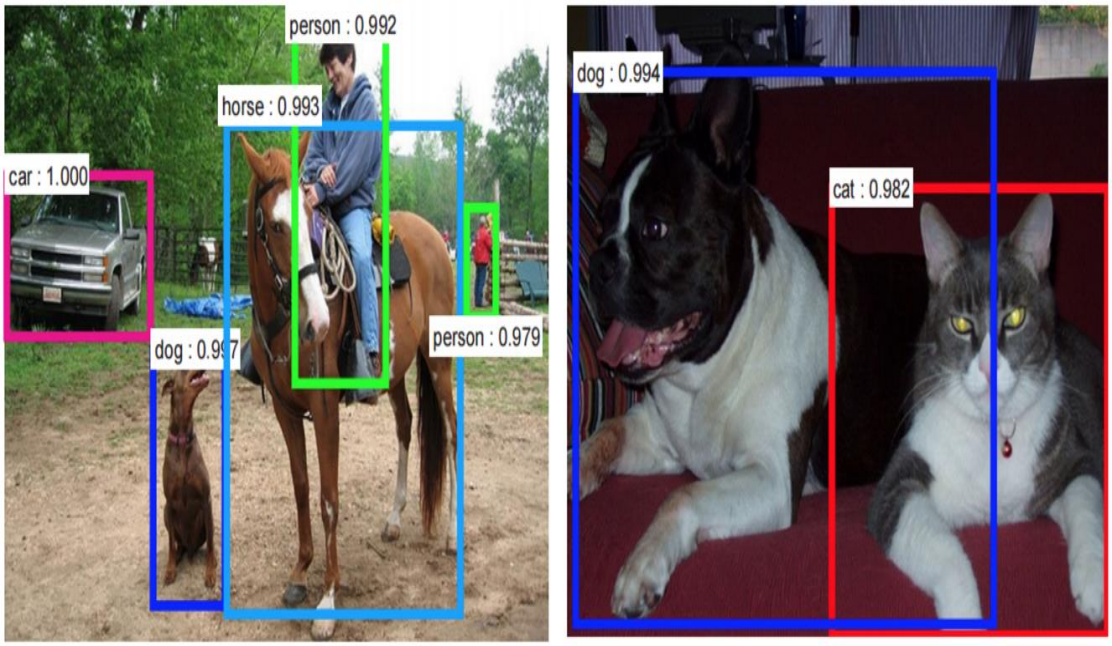


Figure 7

Hence, anatomical neural networks are an important tool for most users of machine learning methods today.[3]

There are **four** basic operations in ConNet in Figure .9.as follows:

i. Convolution or filtering.
ii. Non Linearity (or as it is known using ReLU).
iii. Pooling or Subsampling.
iv. Classification (in the full contact layer) Classification.[9]

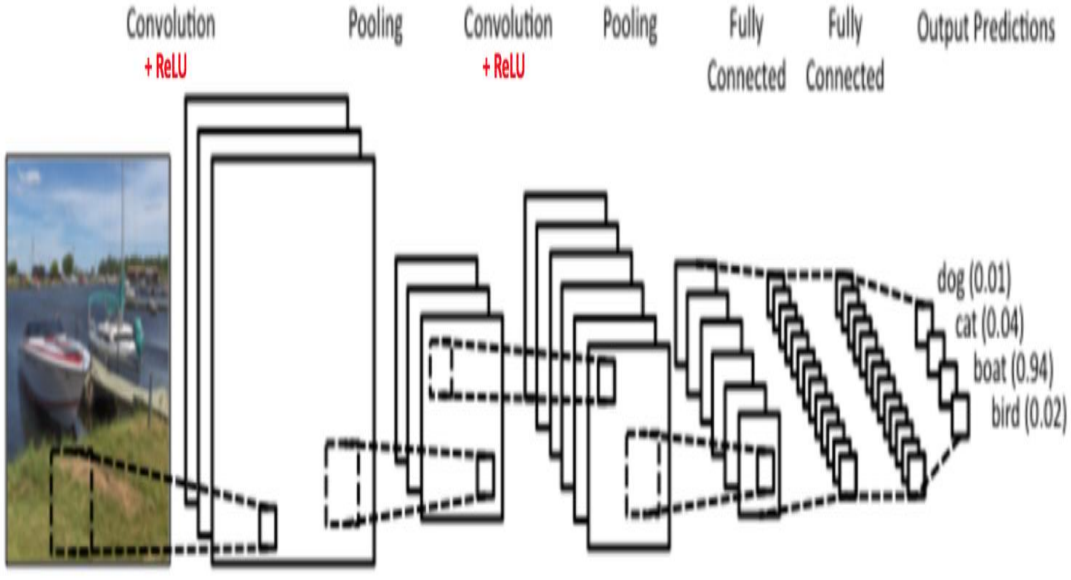


Figure 8

These four processes are the basic component of all anatomical neural networks, and we will explain each step in a simple way to facilitate understanding of the entire process. Before we start, it's important to know how the computer sees the input image. [5]

An image is an array of pixels with specified numerical values.

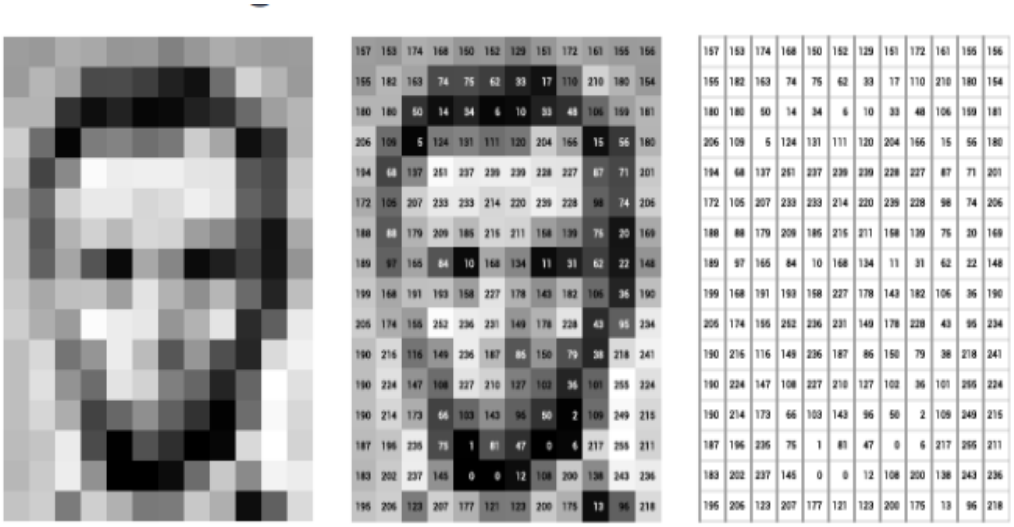


Figure 9 An image is an array of pixels with specific numerical values

i. Convolution or filtration

The primary goal of the convolution process in the case of ConvNet is to extract the properties from the input image so that the wrapping process maintains the spatial relationship between the pixels in the image by learning the image properties using small squares (kernel) on the input data.[12]

We will not explain the mathematical details behind this process but will explain the method for working on the images.

We previously provided detailed anatomy of the image and that it is made up of an array containing pixel values. Now suppose that we have an image consisting of a 5x5 array and the pixel values will be either zero or one for the sake of simplicity (noting that the gray image values range from zero to 255). The following green matrix represents a special case where the pixel values are either zero or one:

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Also, we have assumed another smaller 3x3 array that will serve as a kernel:

1	0	1
0	1	0
1	0	1

What will happen now, as is evident in Figure 11, is that the properties resulting from the convolution process (known as a feature map or a convolved feature) will be extracted by passing the kernel box (yellow matrix) over the green matrix (original image). By one pixel (which is the size of the pass-through step) for each pixel in the image. In each step (Stride), the two matrices will be multiplied together and the result of the multiplication will be combined to extract one value in the properties map (the pink matrix) which will be 3 x 3 in size.

It should be noted that the yellow Kernel box only sees part of the input image in each pass-through step.

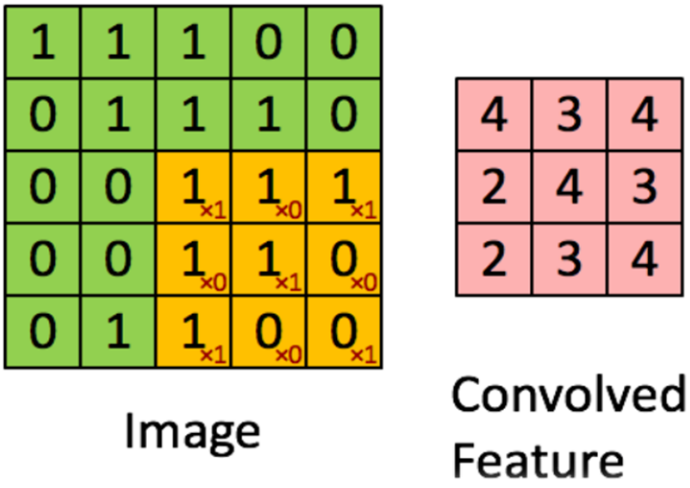


Figure 10 Feature map or Convolved Feature

Now we will detail the concept of the kernel square, which in our current example is a two-dimensional matrix of 3x3 size, usually called a (filter) or (kernel) or (feature detector) and the result of passing this array to the original matrix (the input image) is Also, a new array is known as the "convolved feature", "Activation map", or more commonly called the "feature map".

It is worth noting that these filters act as a feature specifier from the input image.

The reader might think of what values are in the filter matrix or kernels. To clarify, there are different types of filters as each type has a specific effect on the input image and our choice of the filter is according to the project's needs.[4]

Example, let's assume the image below is the input image:

The following table shows the effect of the warping process on the input image using various filters. As shown in the table, we can perform operations like delimiting or blurring the image and so on.

This is done by using different values in the filter matrix. Thus we obtain a map of different characteristics of the image depending on the type of filter used.

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

It is important to know that the wrapping process can determine the local reliability of the pixels (the spatial relationship in the image from the angle of the pixels and all filters are numerical values as previously detailed).

In practice, CNN learns the values of these filters on their own during the training process (considering that we must predetermine some parameters before the model training process such as number of filters, filter size, network structural construction, etc.). The higher the number of filters, the more properties the input image is extracted from, so we have a better grid at identifying patterns in new images that the grid has never seen before.[9]

The size of the characteristic map (wrapped properties) is determined by three basic parameters that must be known before implementing the first step (**the convilution step**):

- **The Depth**

The depth denotes the number of filters used for the warp process, and as is evident in Figure (12), we have implemented the circumvention process on the original image using three different filters.

Thus, we will have three character maps of the input image as if they were three arrays stacked on top of each other.

Thus, the "depth" of the characteristics map becomes three.

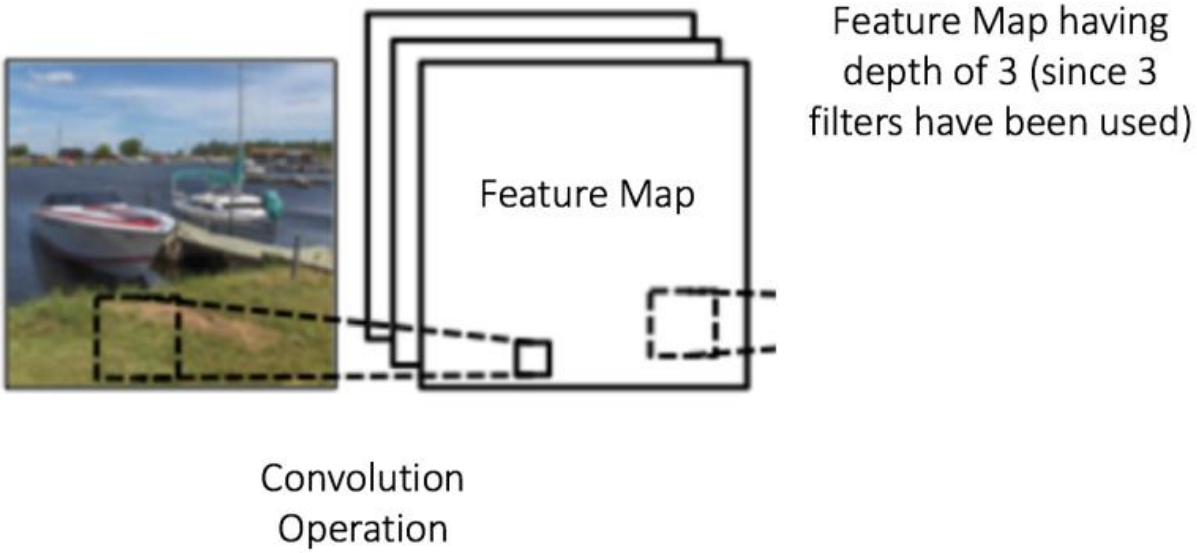


Figure 11

- **Stride pass-through**

The pass-through step is the number of pixels that we will be using when passing the filter matrix over the input image matrix.

When the value of the pass step is one, we move the filter only one pixel at a time, and when the value is 2, the filter jumps two pixels two at a time.

It should be noted that when we have a large size of the pass-through step, this means that we will obtain the characteristic maps of a small size.

- **Zero-padding**

Sometimes it is convincing to line the input image matrix with zeros around the edges of the image and thus we can pass a filter over these edges.

Characteristics extracted from zero padding allow us to control the size of feature maps.

Adding zero padding is also known as a **wide convolution**, and not adding it is known as a **narrow convolution**.[10]

ii. Non Linearity (or as it is known using ReLU)

An additional process called ReLU is used after each convolution in Figure 3 above.

ReLU is an abbreviation for Rectified Linear Unit, and it is a non-linear process whose output is as follows in Figure (13):

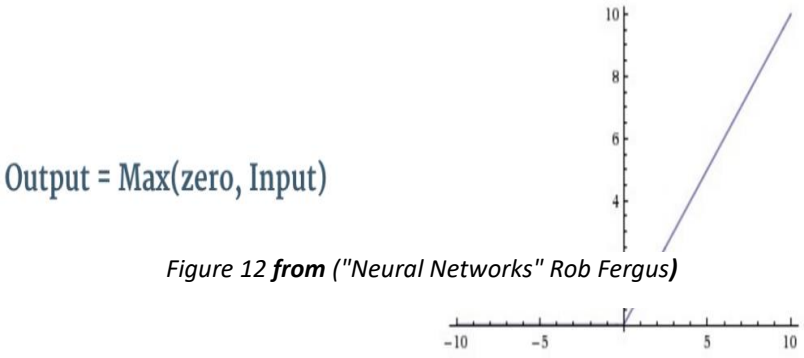


*Figure 12 **from** ("Neural Networks" Rob Fergus**)***

ReLU is an element-level process (applied per pixel) that replaces all negative pixel values in the properties map with a value of zero. The goal is to introduce nonlinearity in ConvNet as most real projects are nonlinear. Basically, convolution is a linear process at the element level in the matrix in terms of multiplication and addition.[12]

Therefore, we use non-linearity by means of ReLU in ConvNet. The ReLU process can be understood more clearly in Figure (14).

The output from applying Relo to the properties map is called the Rectified feature map.

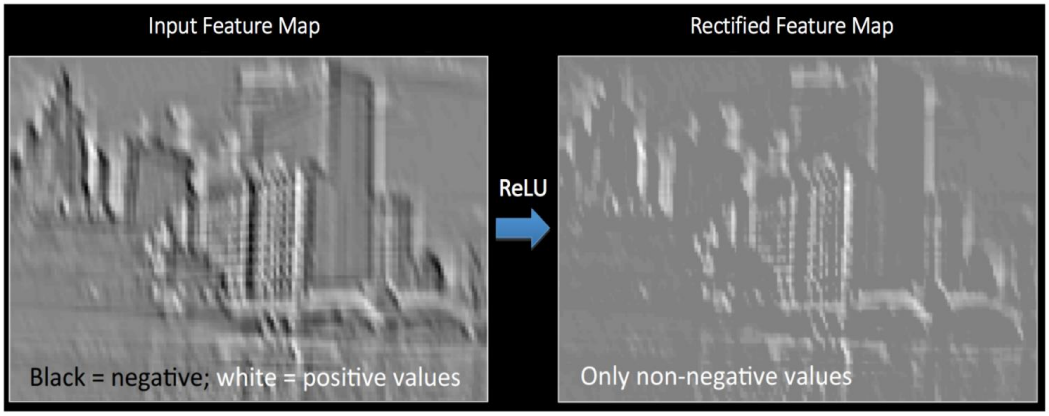


Figure 13

The **ReLU** function is not the only nonlinear function, there are other functions such as **tanh** or **sigmoid**, but **ReLU** has performed better in most cases.[11]

iii.Pooling or Subsampling

Spatial aggregation, also known as subsampling or downsampling, reduces the dimensions of each map's properties while preserving important information.

Spatial aggregation has several types, such as:

- **Max (highest value)**
- **Average (average calculation)**
- **Sum ... etc.**

In the case of grouping using the highest Max Pooling value, we define the contiguous spatial region (a 2x2 window for example) and extract the element (pixel) with the highest value in the specified window from the corrected properties map.

In the same way, if we used the average, we will calculate the average numbers in that window only, or their sum in the case of Sum.

In general, pooling with the highest Max Pooling value showed better performance.

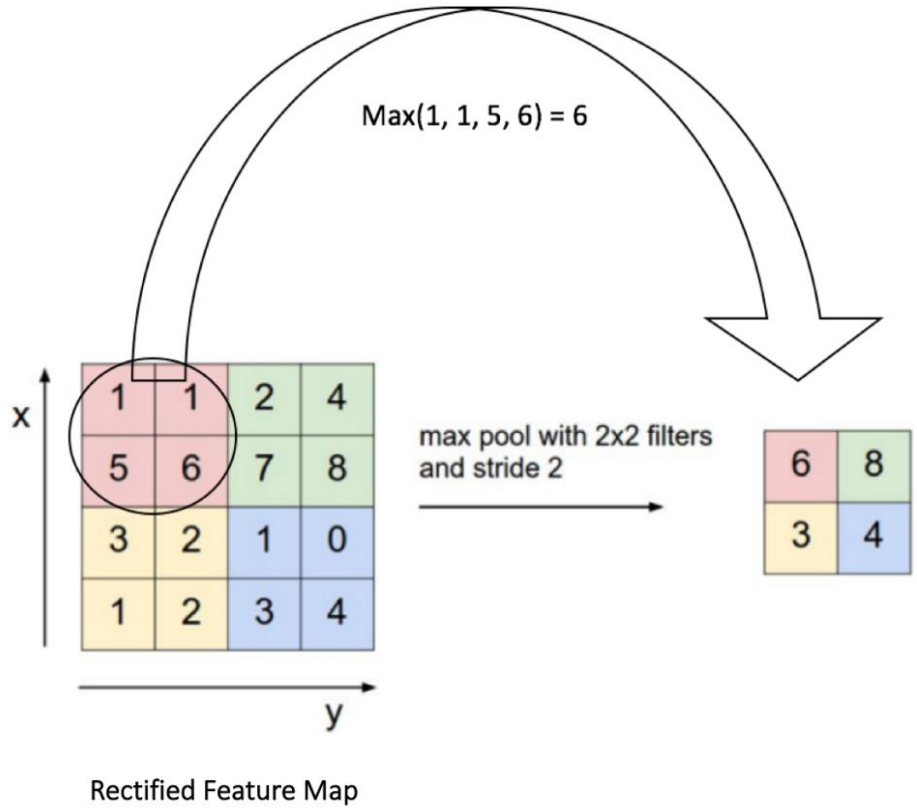


Figure 14

Figure (15) shows an example of using grouping using the highest Max Pooling value on the corrected properties map extracted after the convolution process in addition to the ReLU process using a 2x2 size window

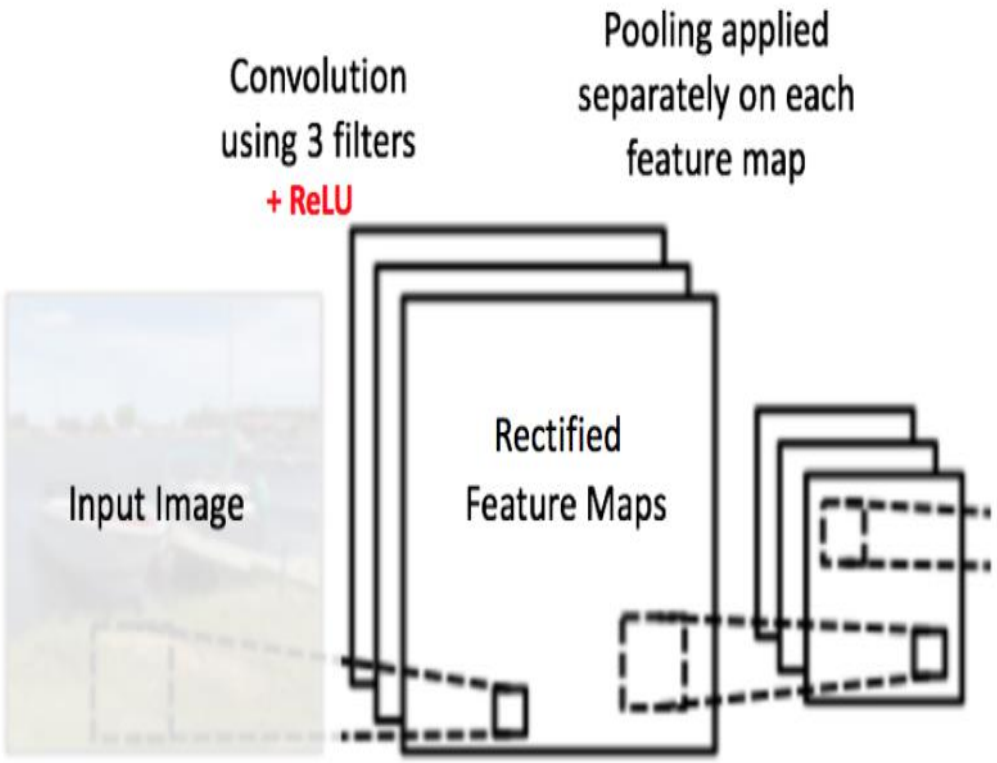


Figure 16

We scrolled a 2x2 window where the stride step has a value of 2 and then we took the highest value for each spatial region.

As noted in Figure (15), this process reduces the dimensions of the characteristic map from 4 x 4 to 2 x 2 in the grid shown in Figure (16).

The pooling process was applied separately to each feature map, and thus we got three maps resulting from three input maps.

The pooling process when applied to the corrected characteristic maps Figure (17) shows the effect of the pooling process on the corrected properties map that we obtained after the ReLU process in Figure (14) above.

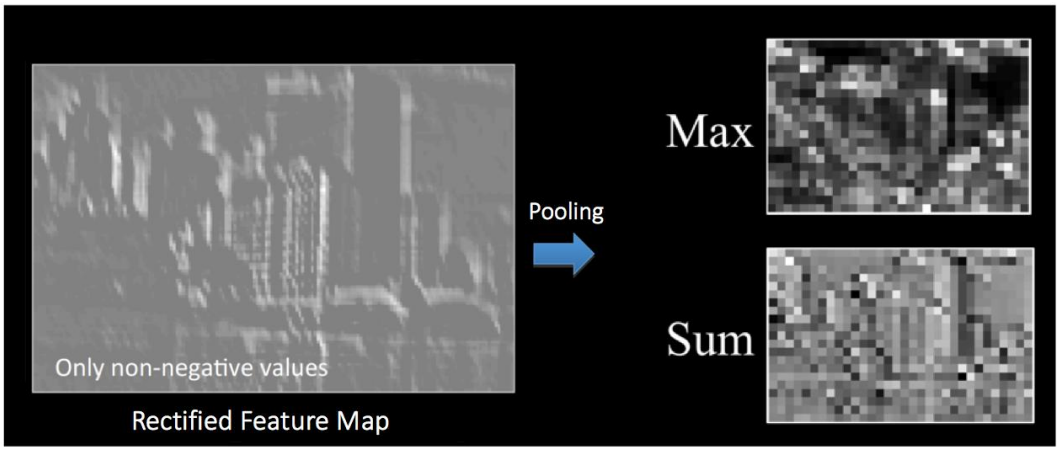


Figure 17 : Pooling

The pooling process continuously reduces the size of the spatial input, and in detail, it does the following:

Makes the representation of the entries in an array of properties smaller in size in terms of the size of the dimensions of the matrix and thus easier to manage.

Reduces the volume of transactions and accounts in the network and thus can control the problem of overfitting

It makes the network stable and resistant to small changes, distortions and shifts in the input image (meaning that a slight distortion in the input will not change the output of the pooling process since we take the maximum / average value in the contiguous local area.

It helps in arriving at a stable and stable representation of the image (the exact term is equivariant) and this is a very powerful feature because it enables us to identify the objects in the image regardless of where they are located.[15]

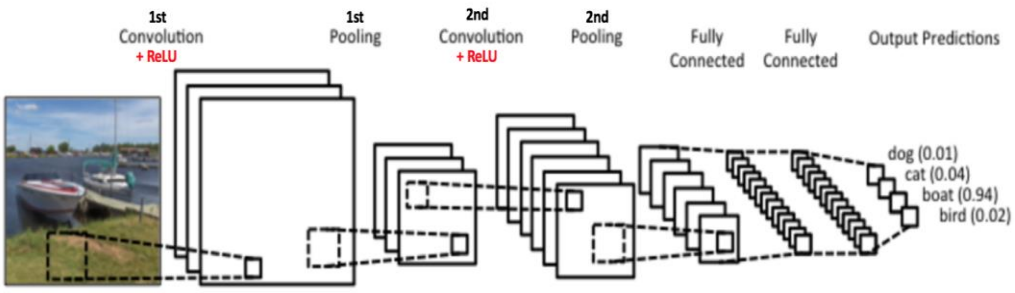


Figure 18

So far we know how **convolution , ReLU, and pooling** work.

It is important to understand that these layers are the basis for building any part of CNN.

As shown in Figure (18), there are two groups of **convolution, ReLU and pooling layers**.

The **second** layer of convolution layers performs the wrapping process on the output of the first **pooling** layer, using 6 filters or filters to extract 6 characteristic maps.

Then the **clustering** process is implemented using the Max Pooling value for each property map separately.

Together, these layers extract useful properties from the input images as well as introduce nonlinearity into the grid and reduce characteristic dimensions while at the same time aiming to make the properties more or less equal in scaling and transformation.

The outputs of the second pooling layer are the inputs to the **Fully Connected** Layer, which will be discussed in the next section.

iV. Classification in the Fully Connected Layer

The fully connected layer is the traditional multi-layer layer that uses the softmax activation function in the output layer (it is possible to use other classifiers in the output layer such as the SVM support machine, but in this article we will stick to the softmax function). The term "fully connected" indicates that every neuron in the previous layer is connected to every neuron in the next layer. It is best to read this article if you do not know what a Multi Layer Perceptron is yet.

The output from the convolution and pooling layers represent high-level characteristics of the input image.

Therefore, the aim of the fully connected layer is to use these properties to classify the input image into several classes based on the training of the data.

For example

The input image classification task that we previously made has four possibilities (dog, cat, boat, bird) as shown in Figure (19) {noting that Figure (19) does not show the connections between neurons in the fully connected layer}.

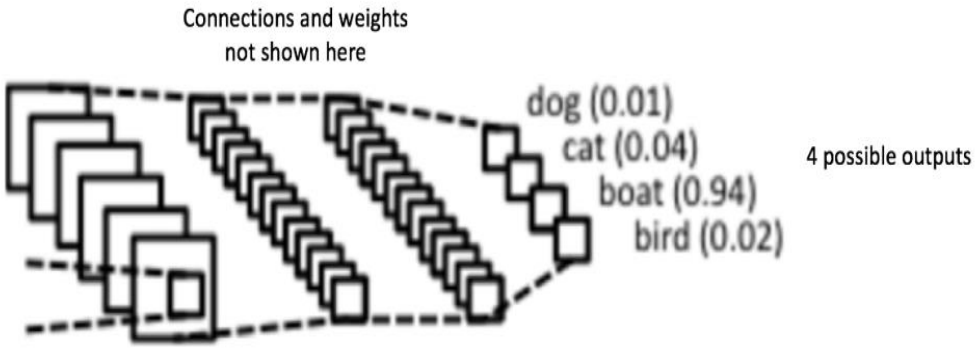


Figure 19

Fully Connected Layer - Each neuron is connected to every neuron in the adjacent layer. Apart from the classification process, the addition of a fully connected layer is usually an inexpensive way to teach nonlinear components of these properties.

Most of the properties extracted from the two layers convolution and pooling may be good for the classification task, but the compounds composed of these properties may be better.

The sum of the output possibilities from the fully connected layer is equal to one, and this is what the softmax activation

function checks for in the output layer in the fully connected layer.

The softmax activation function takes a random vector of real values and then crushes them into a vector with values ranging from zero to one so that their sum is equal to one.

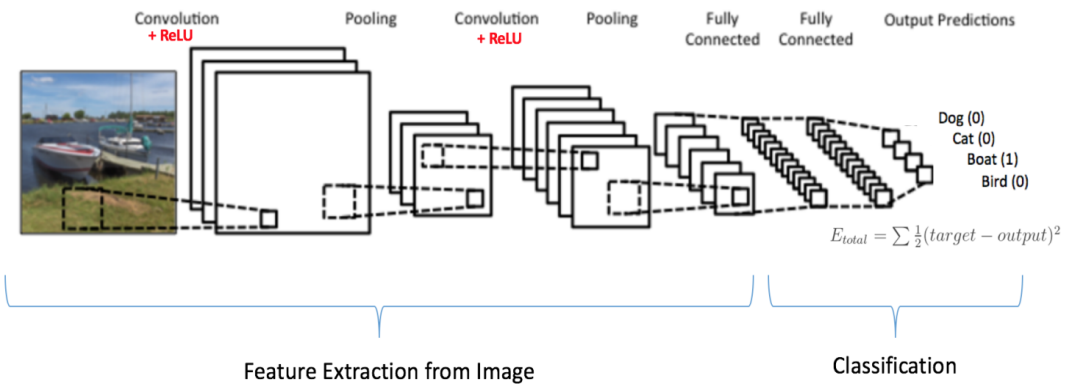


Figure 20

Backpropagation training and use

As explained above, the convolution + pooling layer acts as a feature extractor from the input image, while the fully-connected layer acts as a classifier for the input image using the extracted properties.

Notice in Figure (20) below that the input image is a compound and therefore the probability of the compound class in the output is equal to one while it is equal to zero for the rest of the classes.

Input image = Target vector component = [0,0,1,0]

The overall training process for convolution net can be summarized as follows:

Step 1: We configure all filters and (parameters / weights) with random values

Step 2: The network takes a training image as an input, goes through the forward propagation step (convolution, ReLU, and clustering processes along with forward propagation in the fully connected layer) and looks for the output probabilities for each class.

Suppose the output probabilities of the above boat image are [0.2, 0.4, 0.1, 0.3]

Since weights are randomly assigned to the first training example, the output probabilities are also random.

Step 3: Calculate the total error in the output layer (grouping on all four classes)

Total error = ∑ ½ (target probability - output probability) ²

Step 4: Use Backpropagation to calculate error gradients with respect to all weights in the network and use gradient drop to update all filter values / weights and parameter values to reduce output error.

The weights are adjusted in proportion to their contribution to the total error.

When the same image is entered again, the output probabilities may now be [0.1, 0.1, 0.7, 0.1], which is closer to the target vector [0, 0, 1, 0].

This means that the network learned to classify this particular image correctly by adjusting its weights / filters so that the output error is reduced.[17]

Parameters like number of filters, filter sizes, network structure, etc.are fixed before Step 1 and don't change during the training process - only filter matrix values and connection weights are updated.

Step 5: Repeat steps 2-4 for all pictures in the training set.

The above steps train ConvNet - this basically means that all of the ConvNet weights and parameters are now optimized for correctly classifying images from the training set.

When a new (invisible) image is entered into the ConvNet, the network goes through the forward propagation step and outputs a probability for each category (for a new image, the output probabilities are calculated using weights that have been optimized to correctly classify all previous training examples).

- If our training pool is large enough, (hopefully) the network circulates well on the new images and categorizes them into valid categories.[17]

2.7 What is real time face recogntion?

A real time face recognition system is capable of identifying or verifying a person from a video frame. To recognize the face in a frame, first you need to detect whether the face is present in the frame. If it is present, mark it as a region of interest (ROI), extract the ROI and process it for facial recognition.[1]

Real time face recognition software is divided into two parts:

i. creating a database
ii. training and testing

2.7.i Creating a database

Take pictures of the person for face recognition after running create_database.py script. It automatically creates Train folder in Database folder containing the face to be recognised. You can change the name from Train to the person's name.

While creating the database, the face images must have different expressions, which is why a 0.38-second delay is given in the code for creating the data set. In this example, we take about 45 pictures/images and extract the face, convert it into grayscale and save it to the database folder with its name.[16]

2.7.ii Training and testing

Training and face recognition is done next.

face_rec.py code does everything.

The algorithm used here is Local Binary Patterns Histograms(LBPH).[19]

Figure 21: Screenshot of a LBPH face recogniser

2.8 OpenCV 2.4.10.

OpenCV provides the following three face recognisers:

1. **Eigenface recogniser**
2. **Fisherface recogniser**
3. **LBPH face recogniser**

LBPH face recognition is used, which is create LBPHFaceRecognizer( ) function.

LBP works on gray-scale images. For every pixel in a gray-scale image, a neighbourhood is selected around the current pixel and LBP value is calculated for the pixel using the neighbourhood.

After calculating LBP value of the current pixel, the corresponding pixel location is updated in the LBP mask (it is of same height and width as input image.) with LBP value calculated, as shown in Fig.21.

In the image, there are eight neighbouring pixels. If the current pixel value is greater than or equal to the neighbouring pixel value, the corresponding bit in the binary array is set to 1.

But if the current pixel value is less than the neighbouring pixel value, the corresponding bit in the binary array is set to 0.[15]

2.9 What is pyQt5?

Qt is set of cross-platform C++ libraries that implement high-level APIs for accessing many aspects of modern desktop and mobile systems. These include location and positioning services, multimedia, NFC and Bluetooth connectivity,

a Chromium based web browser, as well as traditional UI development.

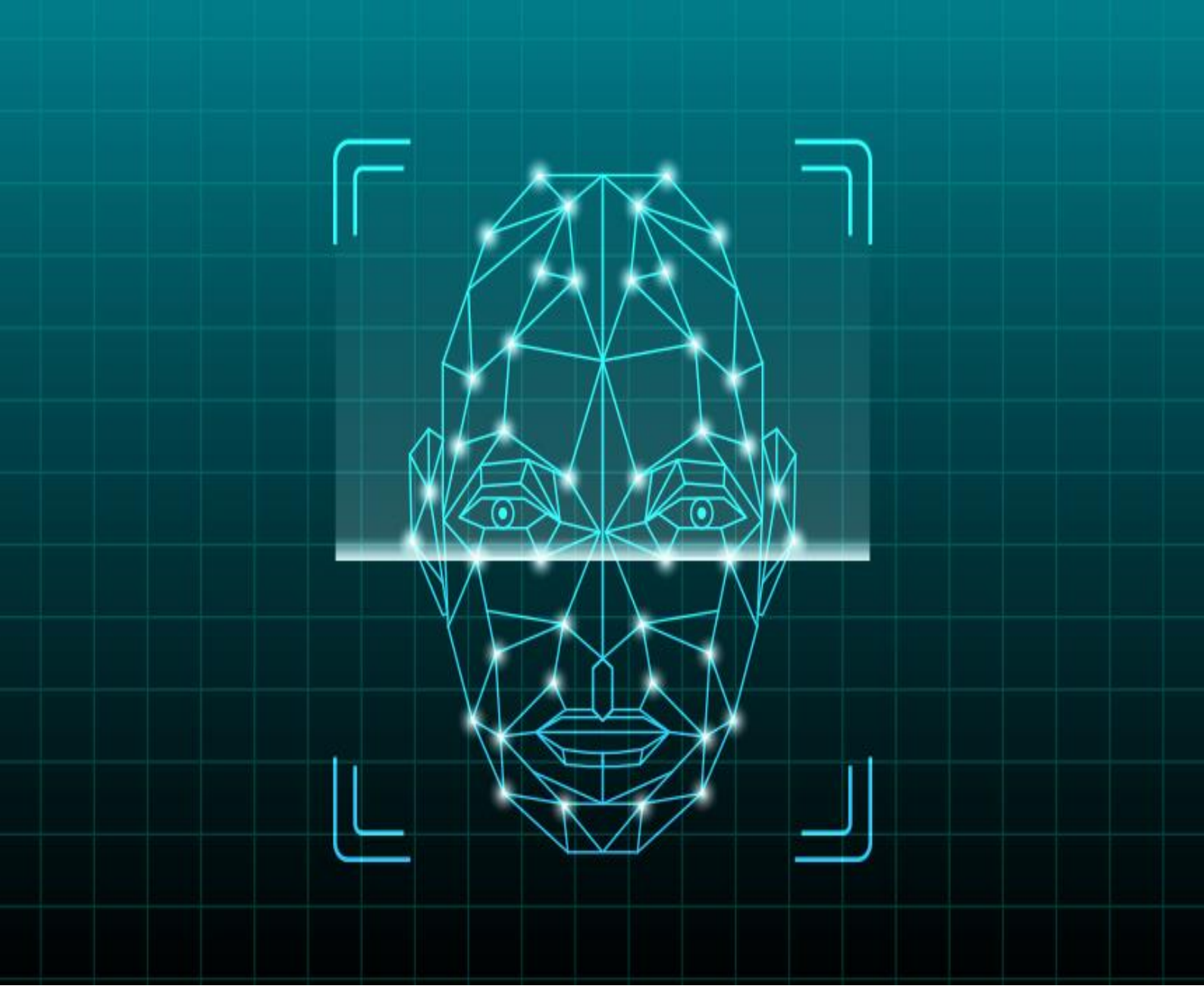
PyQt5 is a comprehensive set of Python bindings for Qt v5. It is implemented as more than 35 extension modules and enables Python to be used as an alternative application development language to C++ on all supported platforms including (Windows, Linux, Mac OS X, Android, iOS).

PyQt5 may also be embedded in C++ based applications to allow users of those applications to configure or enhance the functionality of those applications.

PyQt5 is one of the most used modules in building GUI apps in Python, and that's due to its simplicity as we will see.

As it has many available categories, it has been divided into several units, such as QtCore, QtGui, QtSvg, QtOpengl, QtNetwork, QtXml, and QtSql.

Phil Thompson is the one who creates it.[15]



CHAPTER 3

SYSTEM ANALYSIS

3.1 Introduction

In this chapter we will explain the analysis of the system. This system is all about to make useful an intelligent system by means of facial recognition.

3.2 Purpose

To maintain the attendance record with day to day activities is a challenging task.

The conventional method of calling name of each student is time consuming and there is always a chance of proxy attendance.

The following system is based on face recognition to maintain the attendance record of students.

The daily attendance of students is recorded subject wise which is stored already by the administrator.

As the time for corresponding subject arrives the system automatically starts taking snaps and then apply face detection and recognition technique to the given image and the recognize students are marked as present and their attendance update with corresponding time and subject id.

We have used deep learning techniques to develop this system, used to detect faces in images and deep learning method is used to compute and compare feature facial of students to recognize them.

Our system is capable to identify multiple faces in real time.

3.3 System Requirements Specification

A System Requirements Specification (***SRS***) (also known as a Software Requirements Specification) is a document or set of documentation that describes the features and behavior of a system or software application.

The Software Requirements Specification (***SRS***) is a communication tool between users and software designers.

The specific goals of the ***SRS*** are as follows:

1. Facilitating reviews.

2. Describing the scope of work.

3. Providing a reference to software designers (i.e.navigation aids, document structure)

4. Providing a framework for testing primary and secondary use cases .

5. Including features to customer requirements.

3.3.1 User Class and Characteristics

There are two types of users in this system:

i. ***General users***: can use the application to perform a specific tasks (view weather , translate texts ,..)

ii. ***Administrators***: can perform the same tasks of general users, in addition can modify users database and make reports .

3.3.2 Performance Requirements

Define how well the system performs certain functions under specific conditions. Examples are speed of response, throughput, execution time and storage capacity.

3.4 Flow chart

A flowchart is a diagram that depicts a process, system or computer algorithm.

They are widely used in multiple fields to document, study, plan, improve and communicate often complex processes in clear, easy-to-understand diagrams. Flowcharts, sometimes spelled as flow charts, use rectangles, ovals, diamonds and potentially numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence.

They can range from simple, hand-drawn charts to comprehensive computer-drawn diagrams depicting multiple steps and routes.

If we consider all the various forms of flowcharts, they are one of the most common diagrams on the planet, used by both technical and non-technical people in numerous fields.

Flowcharts are sometimes called by more specialized names such as Process Flowchart, Process Map, Functional Flowchart, Business Process Mapping, Business Process Modeling and Notation (BPMN), or Process Flow Diagram (PFD).

They are related to other popular diagrams, such as Data Flow Diagrams (DFDs) and Unified Modeling Language (UML) Activity Diagrams.

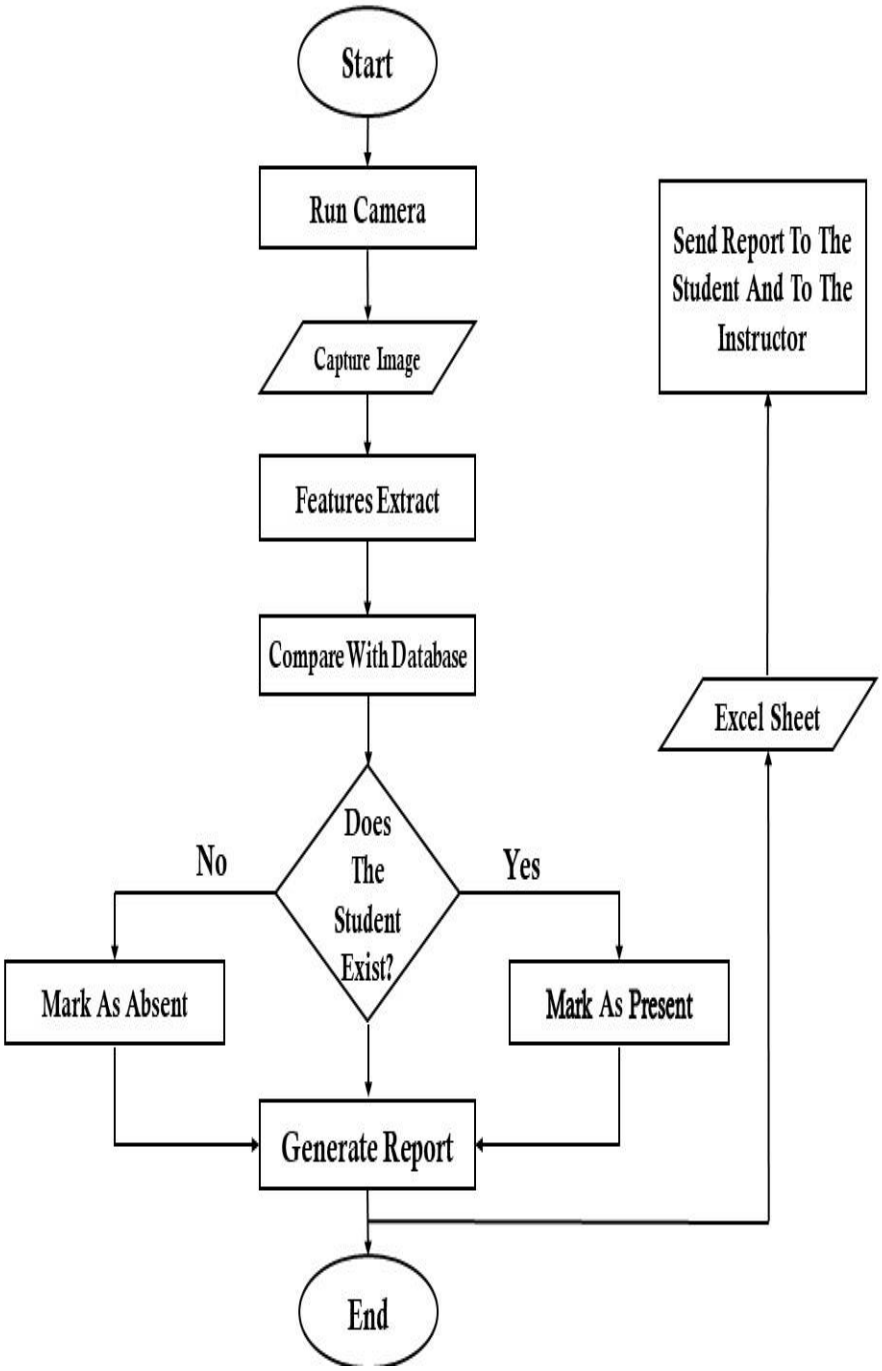


Figure 1 flow chart

3.5 Use Case Diagram

Use case diagram is the primary form of system/software requirements for a new software program underdeveloped.

Use cases specify the expected behavior (what), and not the exact method of making it happen (how).

Use cases once specified can be denoted both textual and visual representation (i.e. use case diagram).

A key concept of use case modeling is that it helps us design a system from the end user's perspective.

It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior.

A use case diagram is usually simple.

It does not show the detail of the use cases:

i. It only summarizes some of the relationships between use cases, actors, and systems.

ii. It does not show the order in which steps are performed to achieve the goals of each use case. As said, a use case diagram should be simple and contains only a few shapes.

If yours contain more than 20 use cases, you are probably misusing use case diagram.

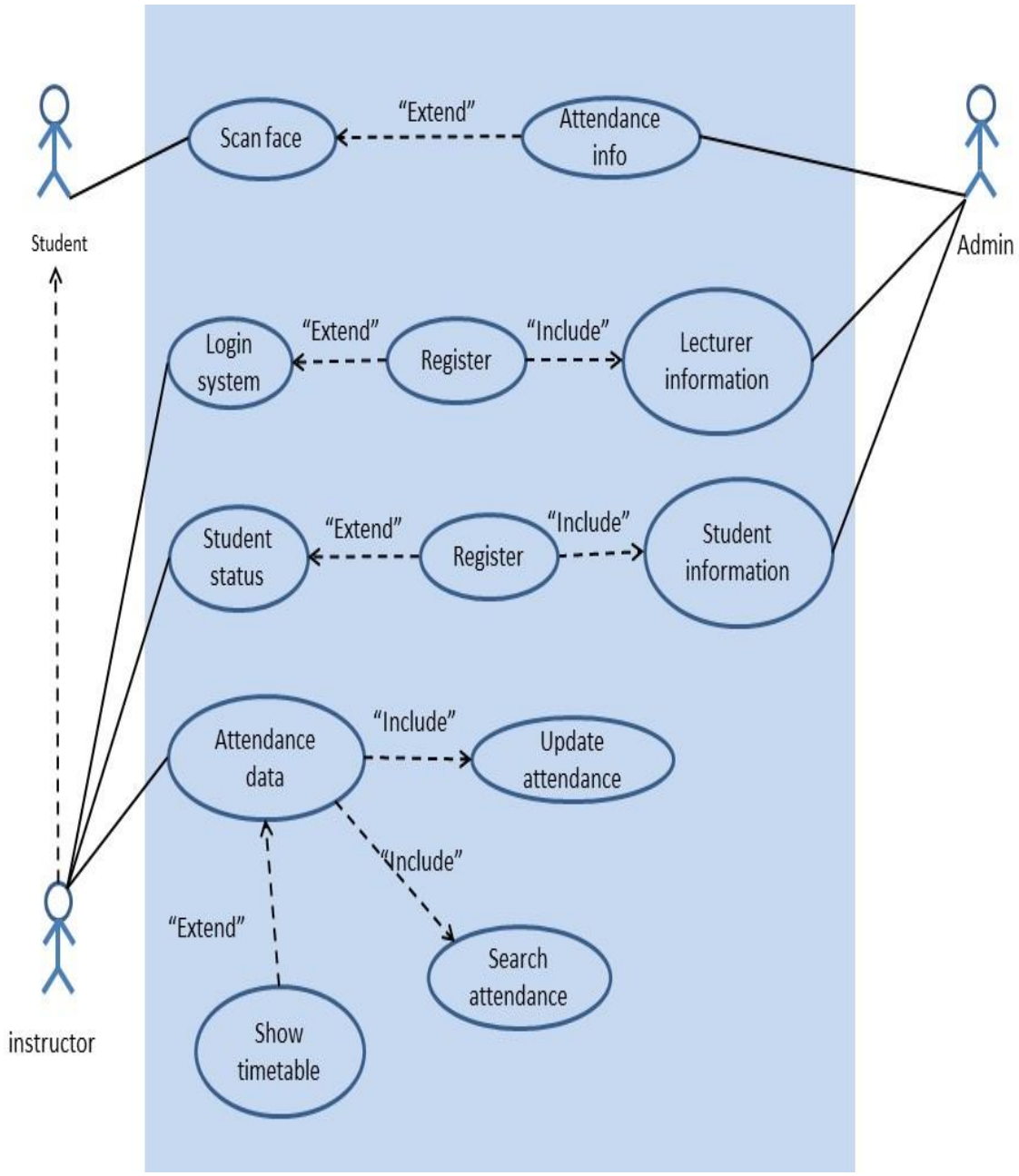


Figure 2 use case diagram

3.6 Sequence Diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place.

We can also use the terms event diagrams or event scenarios to refer to a sequence diagram.

Sequence diagrams describe how and in what order the objects in a system function.

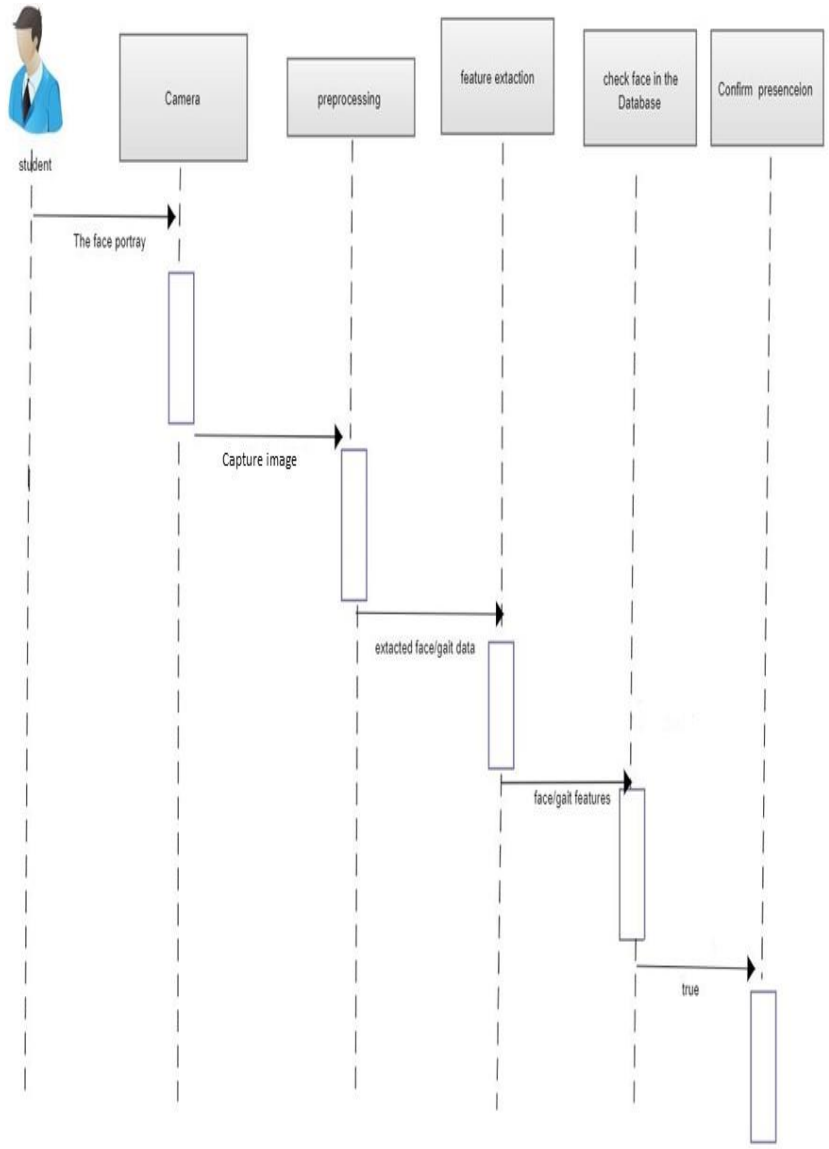
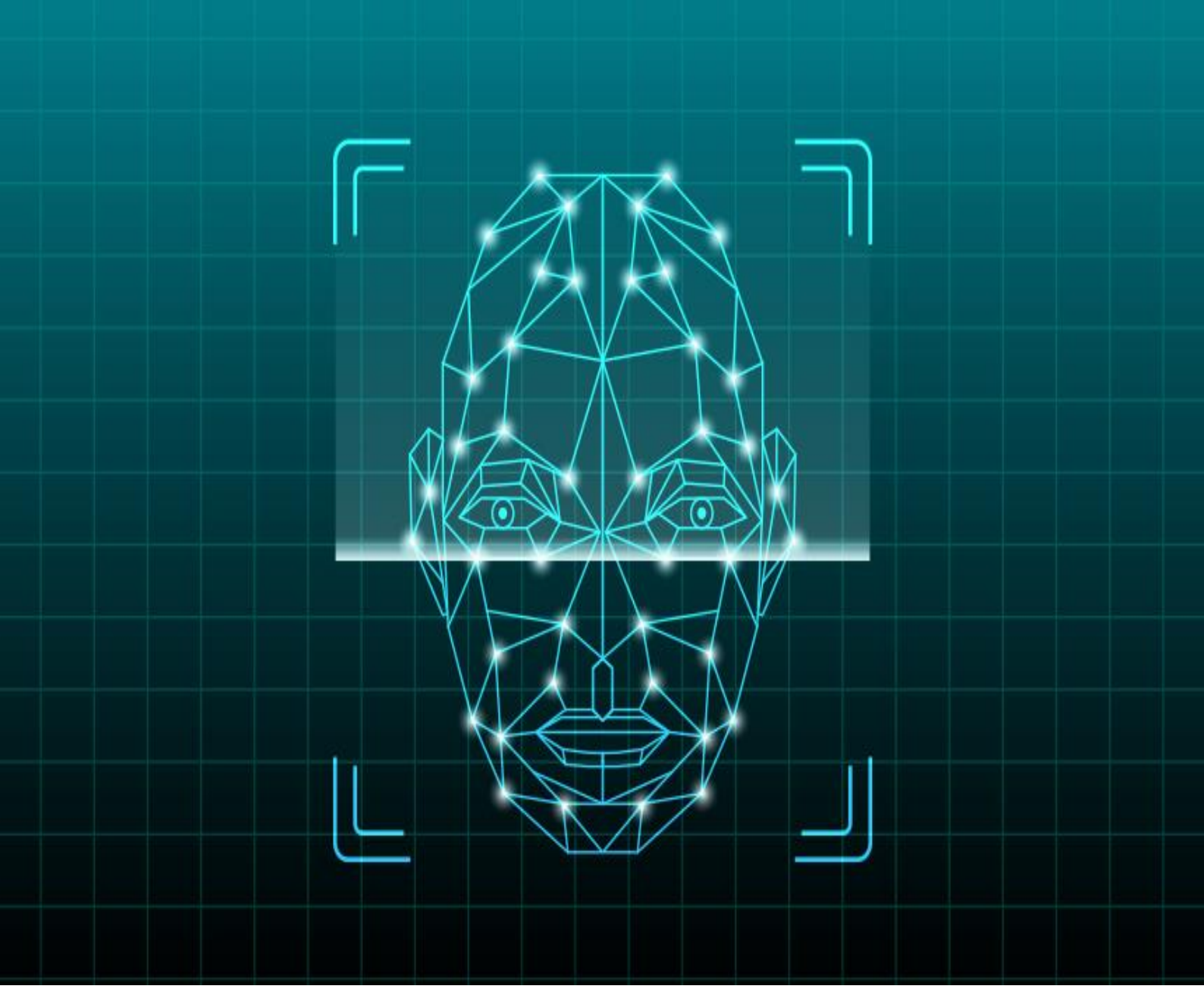


Figure 3 sequence diagram



CHAPTER 4

IMPLEMENTATION

4.1 Introduction

This chapter will contain the implementation of our project, to implement our project after our analysis, first thing to start with implementing the Login Form (figure1) .

4.2 Login Form

Figure 1 Login Form.

4.3 Consisting of:

1. Username -To take name from user.
2. Password -To take password from user.
3. Login button -To check the name and password.
4. Register button -To add a new user.
5. Forgot password -To reset user

4.4 Login button:

If user pressed the Login button, this code will be executed (figure 2).

```
def login(self):
    EMAIL = self.email.text()
    PASSWORD = self.password.text()
    con =mysql.connector.connect(user='root',password='ammar45',host='localhost',database='user')
    manager = con.cursor()
    print('ammar')
    get_email = ("SELECT email FROM users WHERE email = '%s'"%str(EMAIL))
    get_password = ("SELECT pass FROM users WHERE pass = '%s'"%str(PASSWORD))
    manager.execute(get_email,get_password)
    print("welcom")
    if self.email.text()== get_email and self.password.text() == get_password:
        print('ok ok ')
        self.refreshAll()
        ui.hide()  # hide the main window
        self.outputWindow_()  # Create and open new output window
    else:
        print(get_email)
        print(get_password)
        msg = QMessageBox()
        msg.setText("check your email or password")
        msg.exec_()
```

Figure 2 Execute The Code Login Form.

4.5 Register button:

If user pressed the registration button, the registration form will open to add user data (figure 3).

4.6 Registration form:

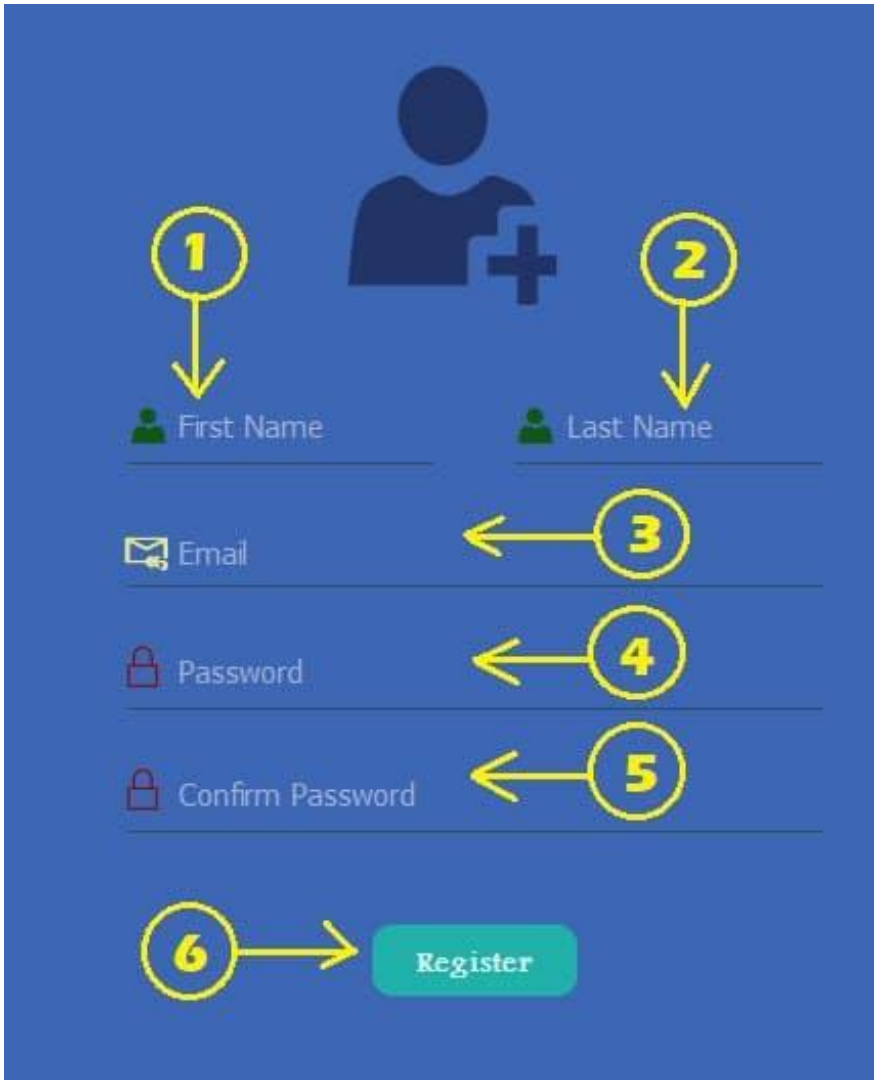


Figure 3 Registration Form.

4.7 Consisting of:

1. First name-This field will take first name from user.
2. Last name-This field will take last name from user.
3. Email- This field will take email from user.
4. Password- This field will take password from user.
5. Confirm password- This field will take confirm password from user.
6. Register button – This button will take all fields and added them in the database.

4.8 Register button:

If user pressed the Register button, this code will be executed (figure 4).

```
def signup(self):
    try:
        con =mysql.connector.connect(user='root',password='ammar45',host='localhost',database='user')
        manager   = con.cursor()
        first_name= self.fname.text()
        last_name =  self.lname.text()
        email     =  self.email.text()
        password1 =  self.pass1.text()
        password2 =  self.pass2.text()
        if (password1 != password2):
            msgBox = QMessageBox()
            msgBox.setText("cheek yor password")
            msgBox.exec_()
        else:
            query = "INSERT INTO users (fname,lname,email, pass) VALUES (%s, %s, %s, %s)"
            value = (first_name,last_name,email, password1)
            manager.execute(query, value)
            con.commit()
            self.fname.setText('')
            self.lname.setText('')
            self.email.setText('')
            self.pass1.setText('')
            self.pass2.setText('')
            msgBox = QMessageBox()
            msgBox.setText("Data Inserted")
            msgBox.exec_()
    except:
        msgBox = QMessageBox()
        msgBox.setText("error")
        msgBox.exec_()
```

Figure 4 Execute The Code Login Form

4.9 Student attendance:

After Login, this form will appear (figure 5).

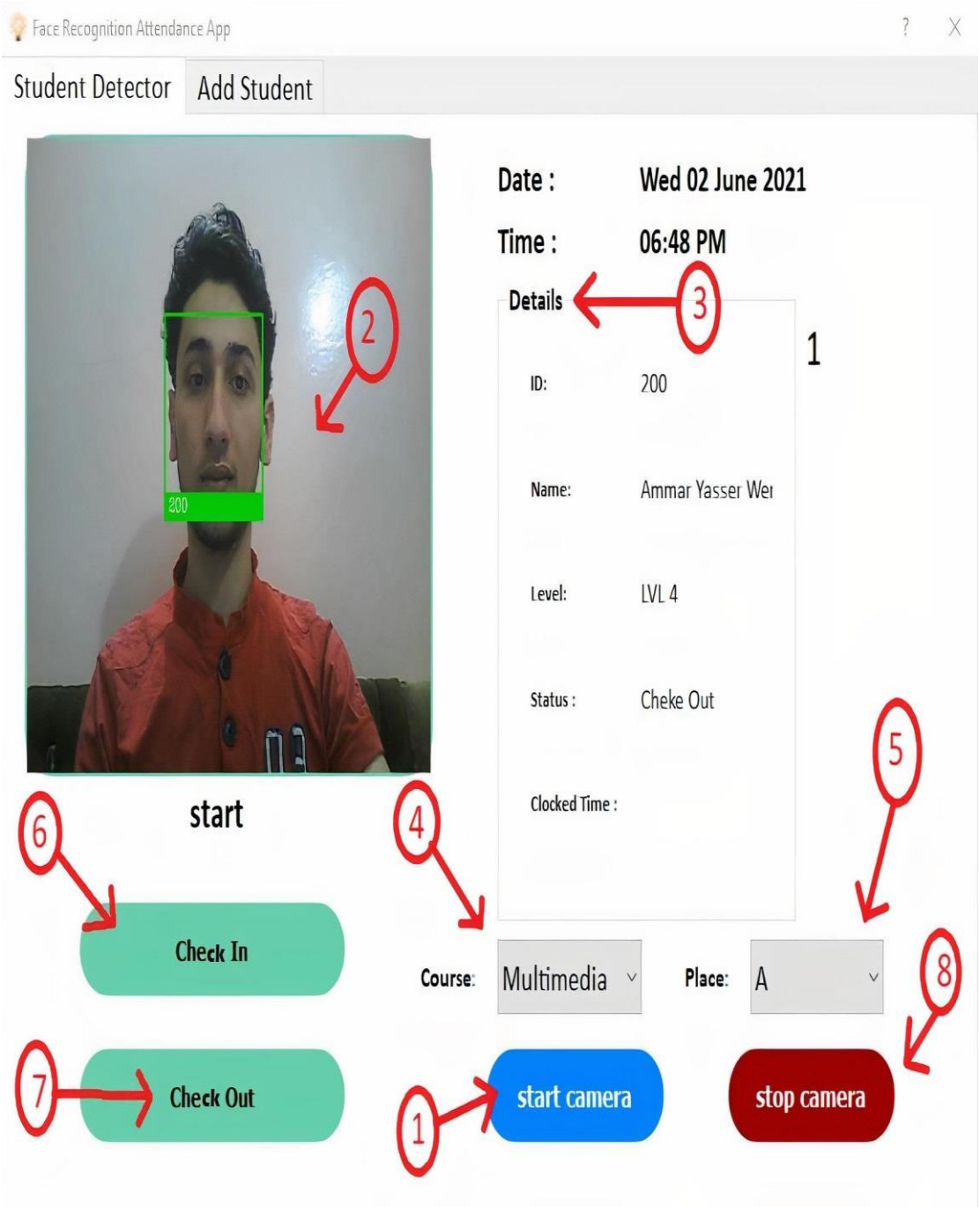


Figure 5 Main Form.

4.10 Consisting of:

1. Start camera button - to open the camera.
2. Form to show the camera and detect face.
3. Student details form - to show details of students.
4. Drop down list for course - to choose the course.
5. Drop down list for place - to choose a place.
6. Check in button - to add all data in the csv file.
7. Check out button - to add all data in the csv file.
8. Stop camera button - to stop camera.

❖ And we added dark mode and white mode for the main form (figure 6).

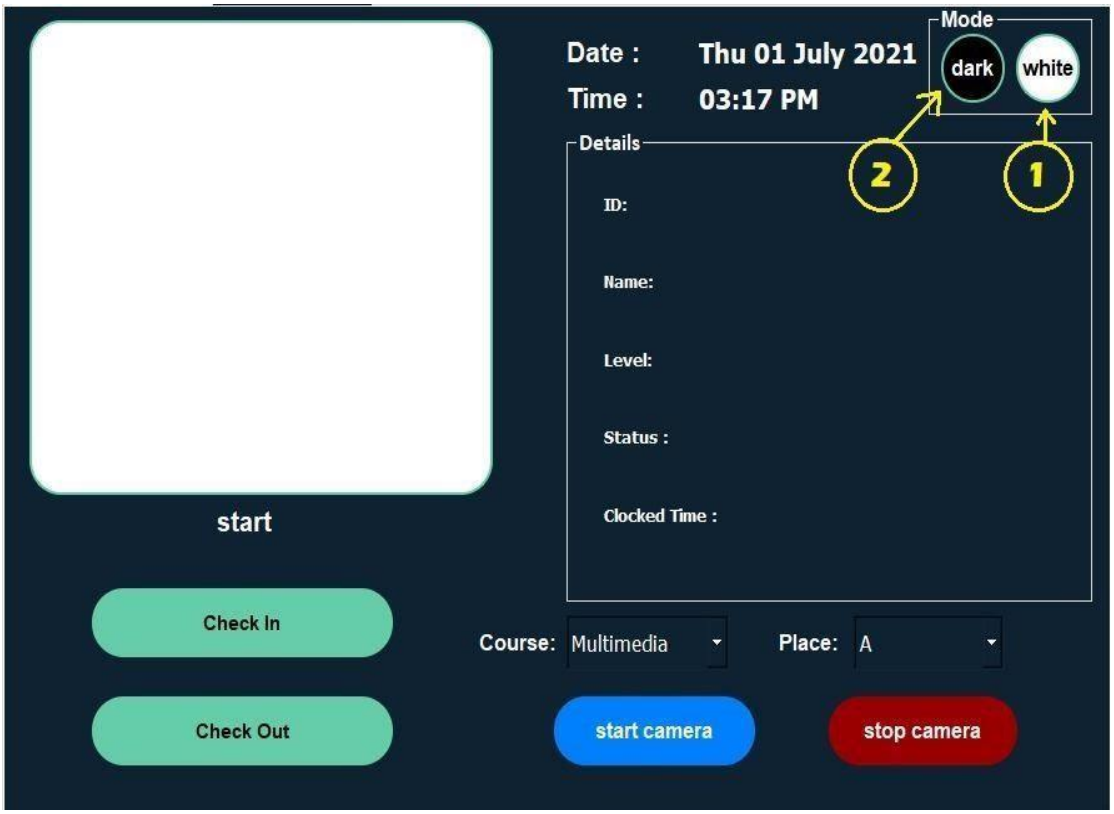


Figure 6 Dark mode form.

❖ **Consisting of:**

- White Mode button.
- Dark Mode button.

1- White mode:

If user pressed the White button, this code will be executed (figure 7).

```
def white(self):
    self.setStyleSheet("background-color: white;color: black")
    self.setWindowTitle("Color")
```

Figure 7 code for white mode.

2-Dark mode:

If user pressed the White button, this code will be executed (figure 8).

```
def dark(self):
    self.setStyleSheet("background-color: black;color: white")
    self.setWindowTitle("Color")
```

Figure 8 code for dark mode.

4.11 Start camera button:

If user pressed the start camera button, this code will be executed (figure 9).

```
def start(self):
        self.NameLabe.setText('start')
        self.capture = cv2.VideoCapture(int(0), cv2.CAP_DSHOW)
        self.checkInButton.setChecked(False)
        self.checkInButton.setEnabled(True)
        path = 'ImagesAttendance'
        if not os.path.exists(path):
            os.mkdir(path)
        images = []
        self.class_names = []
        self.encode_list = []
        self.TimeList1 = []
        self.TimeList2 = []
        attendance_list = os.listdir(path)
        for cl in attendance_list:
            cur_img = cv2.imread(f'{path}/{cl}')
            images.append(cur_img)
            x=cl.rsplit('.', 666)
            self.class_names.append(x[0])
        for img in images:
            try:
                img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
                boxes = face_recognition.face_locations(img)
                encodes_cur_frame = face_recognition.face_encodings(img, boxes)[0]
                self.encode_list.append(encodes_cur_frame)
            except Exception as e:
                print(e)
        if self.NameLabe.text()=='start':
            self.timer.timeout.connect(self.update_frame)  # Connect timeout to the
            self.timer.start(500)  # emit the timeout() signal at x=40ms
```

Figure 9 Start Camera

4.12 Detect Face :

In this point (figures 10,11) will explain how the face recognition system work

```
def face_rec_(self, frame, encode_list_known, class_names):
        def mark_attendance(idimg):
            if self.NameLabel.text()=='':
                text_files = glob.glob( "ImagesAttendance/*.jpg", recursive = True)
                print(text_files)
                print([s.strip('*/') for s in text_files]) # remove the 8 from the string
                x=[s.replace('ImagesAttendance\\', '') for s in text_files]
                print(x)
                if idimg!='unknown':
                    h=idimg
                    final_array = [i for i in range(len(x)) if h in x[i]]
                    d=final_array[0]
                    j=x[d] #name.nans.asdsad
                    print(j)
                    k=j.rsplit('.', 666)
                else:
                    self.NameLabe2.setText('0')
            if (k[1] != 'unknown'):
                self.NameLabe2.setText('1')
                self.IDLabel.setText(idimg)
                self.NameLabel.setText(k[1])
                self.lvlLabel.setText(k[2])
                with open('Attendance.csv', "r") as csv_file:
                    csv_reader = csv.reader(csv_file, delimiter=';')
                    for row in csv_reader:
                        for field in row:
                            self.StatusLabel.setText(field[0:99])
            else:
                self.NameLabe2.setText('0')
        if self.NameLabe.text()=='start':
            faces_cur_frame = face_recognition.face_locations(frame)
            encodes_cur_frame = face_recognition.face_encodings(frame, faces_cur_frame)
```

Figure 10 Execute Code For Detect Face.

```
for encodeFace, faceLoc in zip(encodes_cur_frame, faces_cur_frame):
    match = face_recognition.compare_faces(encode_list_known, encodeFace, tolerance=0.50)
    face_dis = face_recognition.face_distance(encode_list_known, encodeFace)
    idimg = "unknown"
    best_match_index = np.argmin(face_dis)
    y1, x2, y2, x1 = faceLoc
    # print("s",best_match_index)
    if match[best_match_index]:
        idimg = class_names[best_match_index].upper()
        print('kkkkkkkkkkkkkkkkkkkkkkkkkk')
        y1, x2, y2, x1 = faceLoc
        cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 200, 0), 2)
        cv2.rectangle(frame, (x1, y2 - 20), (x2, y2), (0, 200, 0), cv2.FILLED)
        cv2.putText(frame, idimg, (x1 + 6, y2 - 6), cv2.FONT_HERSHEY_COMPLEX, 0.5, (255, 255, 255), 1)

        if  idimg!=self.IDLabel.text():
                self.NameLabe2.setText('')
                self.IDLabel.setText('')
                self.NameLabel.setText('')
                self.lvlLabel.setText('')
                self.StatusLabel.setText('')
                self.HoursLabel.setText('')
                self.MinLabel.setText('')
    mark_attendance(idimg)

return frame
```

Figure 11 Execute Code For Detect Face.

4.13 Check in button:

If user pressed the check in button, this code will be executed(figure 12).

```
def checkIn(self):
        if self.checkInButton.isChecked():
            self.checkInButton.setEnabled(False)
            with open('Attendance.csv', 'a') as f:
                    if (self.NameLabel.text() != '' ):
                        buttonReply = QMessageBox.question(self, 'Welcome ' + self.NameLabel.text(), 'Are you checking In?' ,
                                                           QMessageBox.Yes | QMessageBox.No, QMessageBox.No)
                        if buttonReply == QMessageBox.Yes:
                            text = str(self.comboBox.currentText())
                            text2 = str(self.comboBox2.currentText())
                            date_time_string = datetime.datetime.now().strftime("%d/%m/%y, %H:%M:%S")
                            f.writelines(f'\n{self.NameLabel.text()};{date_time_string};{text};{text2};check In')
                            self.checkInButton.setChecked(False)
                            self.StatusLabel.setText('checked In')
                            self.HoursLabel.setText('Measuring')
                            self.MinLabel.setText('')
                            self.Time1 = datetime.datetime.now()
                            self.checkInButton.setEnabled(True)
                            self.NameLabe2.setText('')
                            self.IDLabel.setText('')
                            self.NameLabel.setText('')
                            self.lvlLabel.setText('')
                            self.HoursLabel.setText('')
                            self.MinLabel.setText('')
                        else:
                            self.checkInButton.setChecked(False)
                            self.checkInButton.setEnabled(True)
                    else:
                        buttonReply = QMessageBox.question(self, 'Welcome ' + self.NameLabe.text(), 'You are unknown!' ,
                                                           QMessageBox.Ok  )
                        self.checkInButton.setChecked(False)
                        self.checkInButton.setEnabled(True)
```

Figure 12 Execute Code for Check In.

4.14 Check out button:

If user pressed check out button, this code will be executed (figure 13).

```
def checkOut(self):
        if self.checkOutButton.isChecked():
            self.checkOutButton.setEnabled(False)
            with open('Attendance.csv', 'a') as f:
                    if (self.NameLabel.text()!= 'unknown'):
                        buttonReply = QMessageBox.question(self, 'Cheers ' + self.NameLabel.text(), 'Are you checking Out?',
                                                           QMessageBox.Yes | QMessageBox.No, QMessageBox.No)
                        if buttonReply == QMessageBox.Yes:
                            text = str(self.comboBox.currentText())

                            text2 = str(self.comboBox2.currentText())
                            date_time_string = datetime.datetime.now().strftime("%d/%m/%y, %H:%M:%S")
                            f.writelines(f'\n{self.NameLabel.text()};{date_time_string};{text};{text2};check Out')
                            self.checkOutButton.setChecked(False)
                            self.StatusLabel.setText('checked Out')
                            self.Time2 = datetime.datetime.now()
                            self.ElapseList(self.NameLabel.text())
                            self.TimeList2.append(datetime.datetime.now())
                            CheckInTime = self.TimeList1[-1]
                            CheckOutTime = self.TimeList2[-1]
                            self.ElapseHours = (CheckOutTime - CheckInTime)
                            self.MinLabel.setText("{:.0f}".format(abs(self.ElapseHours.total_seconds() / 60)%60) + 'm')
                            self.HoursLabel.setText("{:.0f}".format(abs(self.ElapseHours.total_seconds() / 60**2)) + 'h')
                            self.checkOutButton.setEnabled(True)
                        else:
                            print('Not clicked.')
                            self.checkOutButton.setEnabled(True)
```

Figure 13 Execute Code For Check Out.

4.15 Stop camera button:

If user pressed the stop camera button, this code will be executed (figure 14).

```
def stop(self):
    """
    :param camera_name: link of camera or usb camera
    :return:
    """
    self.NameLabe.setText('stop')
    self.imgLabel.setText(" ")
```

Figure 14 Execute Code For Stop Camera.

4.16 Add Student Form:

If user pressed add student button, this form will appear (figure 15).

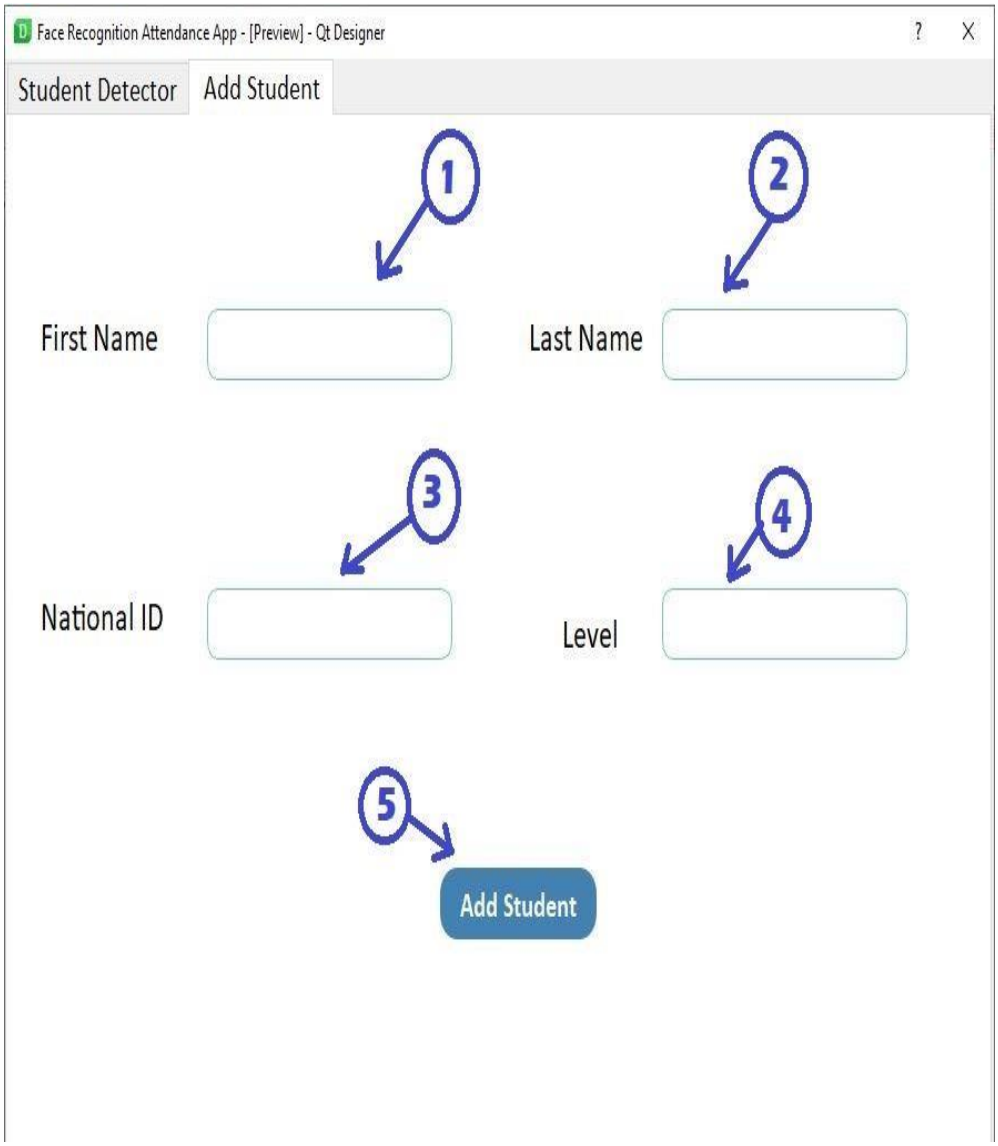


Figure 15 Add Student

4.17 Consisting of:

1- First name to add first name.

2- Last name to add last name.

3- National id to add id.

4- Level to add level.

5- Button to add data in database.

4.18 Add Student button:

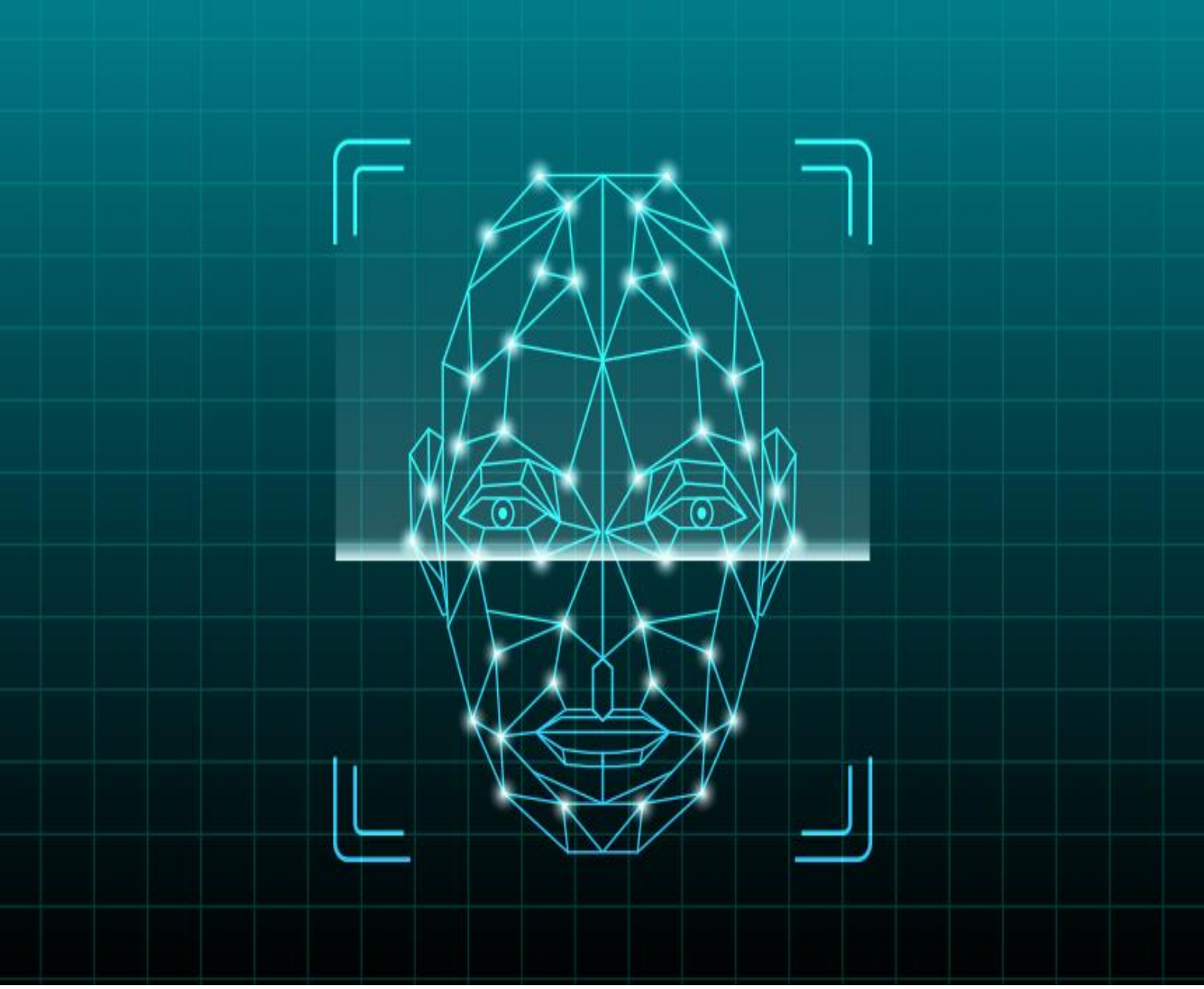
If user pressed the add student button, this code will be executed (figure 16).

```
def addstudent(self):
    try:
        con =mysql.connector.connect(user="root",password="12345",host="localhost",database="student")
        manager   = con.cursor()
        first_name= self.linefname.text()
        last_name =  self.linelname.text()
        n_id      =  self.lineid.text()
        level     =  self.linelevel.text()
        query = "INSERT INTO students (fname,lname,id, levle) VALUES (%s, %s, %s, %s)"
        value = (first_name,last_name,n_id, level)
        manager.execute(query, value)
        con.commit()
        print("Data Inserted")
    except:
        print("Erorr inserting data")
```

Figure 16 Code for add student

To view the source code visit this link:

https://github.com/werdani/face_recognition-system



CHAPTER 5

CONCLUSION AND FUTUER WORK

5.1 Conclusion

The project entitled Building a "**student attendance system**" by python with deep learning completed successfully.

The system has been developed with much care and free of errors at the same time .

the purpose of building this project is that It is important for universities in light of the spread of the new Corona virus [Covid 19].

Because：

- It saves time.
- Reduces congestion.
- It also prevents any error in recording students'attendance.
- Automatic photo capture.

our project was born to make life easier, simpler and more protective

5.2 Future Work

Even after reaching this milestone, we still want to improve Some aspects such as:

i. Increasing the amount of services provided by our system By:

- Enabling the system to send emails, SMS and more.
- Add a function that calculates the maximum student absence and sends a warning SMS to him.

ii. Linking the system to a database that contains the student's schedule of subjects so that the student is not allowed to enter untimely.

References:

1. Shaoqing Ren, *et al,* "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", 2015,arXiv:1506.01497.
2. Eugenio Culurciello's blog , "Neural Network Architectures",2016 https://culurciello.github.io
3. Adam Geitgey ,"machine-learning-is-funpart-3" ,2016,40359318721.
4. Feature extraction using convolution, Stanford Wikipedia article on Kernel (image processing) https://en.wikipedia.org/wiki/Kernel_(image_processing)
5. Joshua Tenenbaum and William Freeman,"Neural Computation", 2000, 12(6):1247-1283.
6. Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, Springer, Heidelberg.
8. https://github.com/karpathy/neuraltalk2

9. Andrew Gibiansky,2014/2/24

URL http://andrew.gibiansky.com/blog/machinelearning/convolutional-neural-networks .

10. Harley, A.W.," An Interactive Node-Link Visualization of Convolutional Neural Networks", Springer International Publishing, Cham (2015) .

11. Lafferty, J.," Advances in Neural Information Processing Systems23", 2011,9781617823800.

12. Vincent Dumoulin, *et al*, "A guide to convolution arithmetic for deep learning", 2015.

13. Nikhil Singh, Paras Ahuja ,"Fundamentals of Deep Learning and Computer Vision",BPB Publications,2020.

14. Kohei Arai, Supriya Kapoor ,"Advances in Computer Vision" 2019,Springer

15. https://en.wikipedia.org/wiki/Artificial_intelligence

16. https://steemit.com/coding/@garethnelsonuk/wtf-areneural-nets.

17. https://techindustan.com/the-contrast-machine-learning-vsdeep-learning/.

18. Asit Kumar Datta, Madhura Datta, and Pradipta Kumar Banerjee," Face Detection and Recognition",2019, 9780367377267.

19. Haowei Liu "Face Detection and Recognition" ,2015, 978-0-12-417045-2.