Wada Nirmiti Education Society

#### *Wada College of Management and Science*

**(Affiliated to University of Mumbai)**

Add: At Harosale, Post Dahe, Tal Wada, Dist Palghar, Pin 421303, Maharashtra (India)

ESTD: 2010

****

**CERTIFICATE**

**This is to certify that \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**(Exam Seat No\_\_\_\_\_\_) of M.Sc. Information Technology class has satisfactory**

**Completed his / her project on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ for the**

**Partial fulfilment of the Degree Master of Science in Information Technology as**

**Prescribed by University of Mumbai.**

**For academic Year 2021-2022.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**External Examiner Principal Professor-in-charge**

|  |  |  |
| --- | --- | --- |
| Sr.No. | Topic | Sign |
|  | Chapter 1 |  |
| 1.1 | Brief Overview |  |
| 1.2 | Face Detection and Recognition |  |
| 1.3 | Motivation |  |
| 1.4 | Problem Definition |  |
| 1.5 | Aim of the Project |  |
|  | Chapter 2 |  |
| 2.1 | Review of Literature |  |
| 2.2 | Comparative Study of review papers |  |
|  | Chapter 3 |  |
| 3.1 | Machine Learning |  |
| 3.2 | Iterative learning |  |
| 3.3 | Approaches to machine learning |  |
| 3.4 | Kernel in CNN |  |
| 3.5 | Why Kernels |  |
| 3.6 | Working of Kernel |  |
| 3.7 | Proposed System |  |
| 3.8 | Haar Cascade |  |
| 3.9 | Local binary patterns histograms (LBPH) Face Recognizer |  |
| 3.10 | Database |  |
| 3.11 | SMTP Protocol |  |
| 3.12 | Email Protocol to Send Alert |  |
| 3.13 | Face Detection Tools |  |
| 3.14 | Language and Library |  |
| 3.15 | UML Diagram |  |
|  | Chapter 4 |  |
| 4.1 | Dataset generation |  |
|  | Chapter 5 |  |
| 5.1 | Conclusion |  |
| 5.2 | Future Scope |  |

# INDEX

# CHAPTER 1

## Introduction

### 1.1 Brief Overview

Nowadays, bunking is preferred by students more often than attending lectures. There are many reasons for a student to bunk lectures. Some students do it under peer pressure some for the fun of it. But it affects their overall academic results and achievements. Most of the time students bunk lectures to hang out with friends. When it comes to attending a lecture or having fun with friends, every student would prefer the later one. They would love to spend time with friends playing basketball, cricket, and going for long bike rides or go to watch a movie. Maybe these memories they will cherish for a lifetime, however, they fail to understand that bunking lectures, to go out with friends can severely impact their grades.

Some students end up bunking lectures to explore college campuses and for networking purposes. Colleges have a big campus equipped with a football court, auditorium, library, and spacious canteen. Students would attend the morning classes and bunk the afternoon sessions to enjoy good food in the canteen, read a novel in the library, play a sport, participate in debates, prepare for the annual function or spend time in extracurricular activities. The extracurricular activity is hosted by colleges for the students to show their talents as annual competitions. Every semester, several competitions are being held in various fields including analytics, programming, debates, public speaking, sports, science, etc. Students who participate in multiple competitions end up bunking lectures due to time clashes.

These are the top reasons why students bunk the class. However, by statistics and expert's information, none of the above reasons is justified for bunking school/ college. All these activities can be done once students finish all the lectures for the day.

Therefore, we have been exploring certain ideas so that students become more disciplined and attend lectures. If there is a system which makes a student accountable for bunking lectures, that will improve the student’s habit of bunking lecture and make them understand the hype is not real for long terms, a system like this would be great and make students responsible for their future.

The process of face recognition is categorized into 2 steps: Face Detection and Face Recognition. The main aim of it is to make the machine aware of what a face looks like. Face detection is primarily governed by few rules like positon of nose, space between eyes and others. Information regarding the face, i.e. position and size of the face from an image or video stream, is collected and noted. However, there are certain issues which are faced during the process of face detection like: Inconvenient circumstances during imaging may lead to problems in distinguishing the foreground from the background. Following the face detection step, the face recognition process can be done. Face recognition is advantageous because of it is simple to perform unlike iris or fingerprint recognition in which the subject’s voluntary participation is required. Depending on geometric information, colour of skin of the subject and other data gathered by any facial recognition algorithm, the system can detect the required face under any lighting conditions. For recognition processes based on skin colour, the image background can interfere with normal processing and recognition of the faces. If in case the face is not correctly extracted from the background of the image, post-processing may be highly affected. Skin colour based face recognition methodologies may be considered advantageous than other methods in that skin colour is not affected by the posture and size in spite of factors like expression changes, moving faces and other such factors.

The Bunk Alert System is not only useful for students but also for parents. It alerts the HOD of the specific student's department if a student is bunking lectures while a teacher is still in the lecture.

This idea is moreover the covering lines on the field of managing the location of students in a college campus and alerting the HOD if a student is found bunking lectures by using the algorithm, of Haar Cascade. This is a machine learning-based approach where a cascade function is trained from a lot of images both positive and negative. Based on the training it is then used to detect the faces or objects in other images. We will be training the machine to identify the faces of the students. We will implement our use case using the Haar Cascade classifier Haar Cascade is a machine learning object detection algorithm used to identify objects or faces in an image or video and based on the concept of ​​ features proposed by Paul Viola and Michael Jones in their paper "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. We have presented an approach for object detection which minimizes computation time while achieving high detection accuracy. LBP Histogram would be used for recognizing the person's ID as well. To implement various examples of computer vision, we are going to use the OpenCV library. OpenCV (Open Source Computer Vision Library) is an open-source BSD-licensed library that includes several hundreds of computer vision algorithms.

Face recognition is the most common way of verification and can also be implemented for communication between machines and humans. Face detection and expression recognition is used widely in industries, corporate sectors, financial exchanges for the purpose of security and authentication. In this paper face detection process uses Haar classifiers and produces an accuracy of around 99.2% from still images and 98.7% for video recordings. Face recognition is done using LBPH algorithm and it produces a recognition accuracy of 99.2% for still images and 99.3% from video recordings for the same have been obtained. Therefore, we will be using LBPH algorithm for our system.

The system identifies a face in live surveillance and then recognizes that person using the LBPH recognizer. The system introduced in this report includes features that are cost, time-effective, and more secure. The system uses already installed surveillance cameras of college authorities which are installed at a secured level from other's reach. So when human presence is detected whenever the surveillance is turned ON as well as the system is turned ON, the system starts capturing student’s faces through the live stream of data. From the frames obtained from the captured video human face is detected and facial features are extracted using the Viola-Jones algorithm and Haar Cascade classifier. The image is compared with the image stored in the dataset as the reference image using the classifier. When the face is recognized an email is sent to the higher authority which here is the department's HOD regarding the captured student.

We can achieve the advantage of image processing and face recognition in our CCTV cameras. Video surveillance and the analysis of the obtained footage is a process that needs huge memory. Video surveillance using CCTV is now being used everywhere. But effective video surveillance is not implemented anywhere. The current practice of video surveillance is installing a camera and analysing the footages manually which are stored. But at the same cost, we can implement something better. Which is alerting about a non-discipline activity in the college campus or department corridor i.e. mass bunking.

### 1.2 Face Detection and Recognition

Computer vision is a field of study which aims at gaining a deep understanding of digital images or videos. Combined with AI and ML techniques, today many industries are investing in researches and solutions of computer vision. Think about the following example: many studies are being carried on to implement security cameras with object detection capabilities. Indeed, imagine a camera in a train station which, depending on the movement captured, can detect whether a fight is occurring. Object detection is a powerful instrument and we will be using it to maintain the attendance rate of students in college using face identification.

Facial recognition technology is the process of identifying or verifying a face from digital images. The need for face recognition has been of real importance with the development of modern society. Detection and recognition of faces have been on the rise worldwide owing to the requirement for security for economic transactions, authorization, national safety and security, and other important factors. The technology comprises face detection, database creation, and face recognition. This system uses a new approach to face identification using the LBPH method and Haar-like features

Face identification is one of the applications of image processing and in this system, it is done by using Haar Cascade feature extraction .xml file. The image processing method is that it will convert an image into the greyscale image and perform some operations on the image, to get an enhanced image or to extract some useful information from it we have to convert it into greyscale since then only we would be able to extract lines and edges from the image.[2] It is a type of signal dispensation in which input is an image, like a video frame or photograph, and output may be an image or associated with that image. Usually, the Image Processing system includes treating images as two-dimensional signals while applying already set signal processing methods to them. Recognizing a face through video stream is done when a human face is first detected and continuous haar features are calculated to identify the face ID match in the existing database. This is done by continuous averaging of white and black pixels of the student's face in the video stream. The process is much faster when the greyscale images are converted to integral images and Adaboost training is done before the recognition process begins.

### 1.3 Motivation

We are final year engineering students and we have known the importance of attending lectures and getting good grades. This project reflects our knowledge of how we preserve college life and how we want it to be. Most of the student's bunk lectures for the fun of and others do it just because the lecture is not interesting enough. And we would love to see this change for our juniors where they don’t feel the need for bunking lectures. This system works in real-time and holds accountable students for bunking lectures and if the organization wants to see the class full of students they would conduct lectures in many effective ways. Like make lectures interesting through visuals, stories, and language that connects with students, use practical aspects to teach students or ask students that what makes learning interesting for them. It could be VR or hands-on projects since engineering students always like to get their hands dirty. But here comes the important aspect that if a student is not present in the class then for whom a teacher would make the learning process more interesting. This project will improve the overall presentation of the student being present in the lectures. In turn, it will help them to inculcate good habits of learning in class. This Bunk Alert System will update the HOD if a student is captured while roaming in the corridors. Due to which the HOD can take account of it and ask the student to meet them in their office. It will surely secure the student’s academics and life.

### 1.4 Problem Definition

The system under consideration is a bunk alert system, which is going to be used in college premises to identify if a student of any department is roaming in their department corridor during lecture hours. There are many times that a student doesn’t attend lecture just because he/she wants to hang out with other students in college canteen or gym which affect a student academic marks, which in turn put a student’s future in a bad condition. Students neglecting to attend a lecture can have various reasons but they have not justified reasons most often. Therefore, this system is designed to keep a student accountable for the attendance and duties of a responsible student. Since this project is made by college students itself who are from the IT department it is feasible to use their field knowledge to overcome the problem by using all of the reasonable ideas which will be cost-effective and don’t require much computer resources. A CCTV (closed-circuit television) system allows the use of video cameras to monitor the interior and exterior of a property, transmitting the signal to a monitor or set of monitors. More and more of us are switching on the benefits of a CCTV security system. We have chosen the concept of computer vision with machine learning. Since an organization always has a surveillance system and some computers with powerful processing speed this project can be adapted by any organization.

### 1.5 Aim of the Project

The overall aim of surveillance is to provide information that is required to inform public health action. Although 'information for action' is often used as a meaningful synonym for surveillance, it does not help to define the purpose of surveillance any further unless we can answer the questions of 'what action?', 'what information?' and 'whose action?'

With the presence of CCTV cameras, discipline can be easily achieved. Whenever human minds are aware that they are being monitored throughout, they easily deter from engaging in hooliganism. It is the responsibility of the school authorities to strengthen the security in school premises with the string of security cameras.

Bunk Alert System is designed to reduce the lecture bunking rate of college students and making them accountable to their lecture’s attendance rate because this system is transparent and works in real-time. This system is built by using trending computer vision field’s machine learning algorithm which can recognize faces in a video frame by frame and extract that person’s facial features using a haar cascade which will further send it to LBPH recognizer, the person by sorting through SQLite table in database and students face id with haar features. And finding the correct match. If the person is in the system’s database and is roaming in college premises during lecture hours then an alert email is sent to that department's HOD with the student’s id, name, and gender and then the HOD can take further action upon it. If the person identified in the CCTV video frame is not present in the system's database then the system will not send an alert to anyone.

**CHAPTER 2**

## 2.1 Review of Literature

This project is moreover the covering lines in the field of managing the location of students in a college campus and making them attend lectures by using this algorithm, Face detection and recognition has become one of the most popular topics of research in the recent times as it is one of the suitable and reliable ways of accrediting a person’s identity. This aspect may be implemented in security and biometric identification methodologies. Over the years various methods have been put forward to accomplish the task of face recognition which includes the use of neural networks to identify faces, elastic template matching, algebraic moments, lines, and others. The process of face recognition is categorized into 2 steps: Face Detection and Face Recognition. The main aim of it is to make the machine aware of what a face looks like. Face detection is primarily governed by few rules like the position of the nose, space between eyes, and others. Information regarding the face, i.e. position and size of the face from an image or video stream, is collected and noted. However, certain issues are faced during the process of face detection like inconvenient circumstances during imaging may lead to problems in distinguishing the foreground from the background. A solution to this issue may be addressed by using larger and better training samples while training a good template.

The paper by Shalini SinghMd, Golam Mohiuddin, Swati Pandit, Amogh Banerjee, Indrasom Gangopadhyay, Indrajit Das called as **“Face Detection and Recognition by Haar Cascade Classifier, Eigen Face and LBP Histogram”** has provided us by information about face recognition using Haar Cascade, Eigen Face and LBPH (Local binary patterns histograms) and their accuracy with still images and live video frames and other aspects as follows.

 Face recognition is the most common way of verification and can also be implemented for communication between machines and humans. Face detection and expression recognition are used widely in industries, corporate sectors, financial exchanges for security, and authentication. In this paper, the face detection process uses Haar-classifiers and produces an accuracy of around 99.2% from still images and 98.7% for video recordings. Face recognition is done using 2 algorithms i.e. LBPH algorithm and Eigenfaces respectively former produces a recognition accuracy of 99.2% for still images and 99.3% from video recordings and latter produces an accuracy of 98.8% and 99.1% for the same have been obtained.

The paper by Paul Viola and Michael Jones named as **“Rapid Object Detection using a Boosted Cascade of Simple Features”**, also describes a machine learning approach for visual object detection which is capable of processing images extremely rapidly and achieving high detection rates. This work is distinguished by three key contributions. The first is the introduction of a new image representation called the “Integral Image” which allows the features used by our detector to be computed very quickly. The second is a learning algorithm, based on AdaBoost, which selects a small number of critical visual features from a larger set and yields extremely efficient classifiers. The third contribution is a method for combining increasingly more complex classifiers in a “cascade” which allows background regions of the image to be quickly discarded while spending more computation on promising object-like regions. The cascade can be viewed as an object-specific focus-of-attention mechanism which unlike previous approaches provides statistical guarantees that discarded regions are unlikely to contain the object of interest. In the domain of face detection, the system yields detection rates comparable to the best previous systems. Used in real-time applications, the detector runs at 15 frames per second without resorting to image differencing or skin color detection.

2.1.1 Rowley et al (1998) have provided a neural network-based upright frontal face detection system in “Neural Network-Based Face Detection”. To collect negative examples, a bootstrap algorithm is used, which adds false detections into the training set, as training progresses.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Author** | **Year** | **Technique.** | **Pros.** | **Cons.** |
| Steve Lawrence C.Lee Giles Andrew D Back | 1997 | Convolutional neural network | It provides partial invariance to translation, rotation, scale, and deformation.96.2  % error. | Computation complexity. Error of 3.8% . |
| Paul viola, Michael Jones | 2001 | The boosted cascade of simple features. | It detects objects from  region, not background using line and edge strips diagram. 99% is it’s accurate. | It detects without restore to image difference and skin color detection.1  %error is detected. |
| Florian Schroff, Dmitry Kalenich enko, James Philbin | 2015 | A unified embedding for face recognition and clustering | It gives greater representation efficiency.  It can be used only for 128 bytes per face. | There is an error that occurred regarding datasets. |
| Tabas Baltrusait is, Peter Robinson  ,  Louis Philippe Morency | 2016 | Open source facial. Behavior analysis toolkit | It is capable of facial landmark detection, head pose estimation, facial action unit recognition, and eye gaze  estimation. | It cannot detect the face if there are not specific features like a facial landmark. |
| Wesley  L. Passos Igor M. Quintanil ha Gabriel M. Araujo | 2018 | Real-time DL based | It is useful in fiducial points extraction and embedding.  It is also useful for the classification task. 99.9% accuracy. | Showing error in facial features detection and face recognition.  0.123% error is detected. |
| Bienveni do, bartido, abad jr | 2018 | Image processing technique | Identification rate is high with threshold measurement | Normal and illustration normalization should be observed when setting a  threshold to avoid false identification. |
| Chung Hua Chu, Yu Kai Feng | 2018 | Polynomial neural network classifier on data preprocessing technique | Authentication by eye blink (non-biometric detection), 99% is accurate. | Computational complexity. |
| I. Gallo, S. Nawaz, A. Calefati | 2018 | Convolutional neural network using pipeline | Generic pipeline is capable of creating, cleaning and recognition from videos and images.99.33% accuracy  obtained. | Video with low pose variability not detect well. 0.66% of error is detected. |
| Dr. Priya Gupta, Nidhi Saxena, Meetika Sharma, Jagriti Tripathi | 2018 | Deep neural network and CovNet technique | 97.05 %  accuracy is obtained. | 2.95% error is detected. |
| Muhamm ad Imran Razzak, Saeeda Naz ,and Ahmad Zaib | 2019 | Deep learning for image processing | It is used for medical image segmentation and classification basically in health sector. | It is limited in subjectivity, the complexity of the image, extensive variations. |

# CHAPTER 3

## System Architecture and Design

The system consists of various procedures to process the system inefficient manner. Many of them are to detect and identify faces but behind the scene, various algorithms are working to make the system function properly, for example, Haar Cascade, Adaboost and LBP histogram algorithm, etc. The technology comprises face detection, database creation, and face recognition. This system uses a new approach to face identification using the LBP method and Haar-like features. The first step is face detection which is done using the Haar cascade classifier. After detection, a face is saved in the database. Then the faces from the database are passed through the face recognition algorithm. We will be looking over each of them in detail, in this chapter.

### Machine Learning

Machine learning is a form of AI that enables a system to learn from data rather than through explicit programming. However, machine learning is not a simple process. As the algorithms ingest training data, it is then possible to produce more precise models based on that data. A machine-learning model is the output generated when you train your machine-learning algorithm with data. After training, when you provide a model with an input, you will be given an output. For example, a predictive algorithm will create a predictive model. Then, when you provide the predictive model with data, you will receive a prediction based on the data that trained the model.

### Iterative learning

Machine learning enables models to train on data sets before being deployed. Some machine- learning models are online and continuous. This iterative process of online models leads to an improvement in the types of associations made between data elements. Due to their complexity and size, these patterns and associations could have easily been overlooked by human observation. After a model has been trained, it can be used in real time to learn from data. The improvements in accuracy are a result of the training process and automation that are part of machine learning.

### 3.3 Approaches to machine learning

Machine-learning techniques are required to improve the accuracy of predictive models. Depending on the nature of the business problem being addressed, there are different approaches based on the type and volume of the data. In this section, we discuss the categories of machine learning.

### 3.3.1 Supervised learning

Supervised learning typically begins with an established set of data and a certain understanding of how that data is classified. Supervised learning is intended to find patterns in data that can be applied to an analytics process. This data has labeled features that define the meaning of data. For example, you can create a machine-learning application that distinguishes between millions of animals, based on images and written descriptions.

### 3.3.2 Unsupervised learning

Unsupervised learning is used when the problem requires a massive amount of unlabeled data. For example, social media applications, such as Twitter, Instagram and Snapchat, all have large amounts of unlabeled data. Understanding the meaning behind this data requires algorithms that classify the data based on the patterns or clusters it finds. Unsupervised learning conducts an iterative process, analyzing data without human intervention. It is used with email spam-detecting technology. There are far too many variables in legitimate and spam emails for an analyst to tag unsolicited bulk email. Instead, machine-learning classifiers, based on clustering and association, are applied to identify unwanted email.

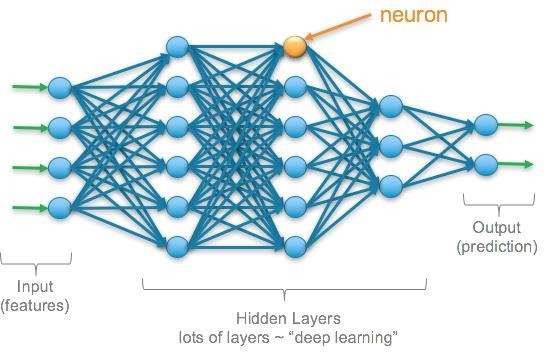
### 3.3.3 Reinforcement learning

Reinforcement learning is a behavioral learning model. The algorithm receives feedback from the data analysis, guiding the user to the best outcome. Reinforcement learning differs from other types of supervised learning, because the system isn’t trained with the sample data set. Rather, the system learns through trial and error. Therefore, a sequence of successful decisions will result in the process being reinforced, because it best solves the problem at hand.

### 3.3.4 Deep learning

Deep learning is a specific method of machine learning that incorporates neural networks in successive layers to learn from data in an iterative manner. Deep learning is especially useful when you’re trying to learn patterns from unstructured data. Deep learning complex neural networks are designed to emulate how the human brain works, so computers can be trained to deal with poorly defined abstractions and problems. The average five-year-old child can easily recognize the difference between his teacher’s face and the face of the crossing guard. In contrast, the computer must do a lot of work to figure out who is who. Neural networks and deep learning are often used in image recognition, speech, and computer vision applications.

Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader family of [machine learning](https://en.wikipedia.org/wiki/Machine_learning) methods based on artificial neural networks. Learning can be [supervised,](https://en.wikipedia.org/wiki/Supervised_learning) [semi-supervised](https://en.wikipedia.org/wiki/Semi-supervised_learning) or [unsupervised.](https://en.wikipedia.org/wiki/Unsupervised_learning) Deep learning architectures such as [deep neural networks](https://en.wikipedia.org/wiki/Deep_learning#Deep_neural_networks), [deep belief networks](https://en.wikipedia.org/wiki/Deep_belief_network), [recurrent](https://en.wikipedia.org/wiki/Recurrent_neural_networks) [neural networks](https://en.wikipedia.org/wiki/Recurrent_neural_networks) and [convolutional neural networks](https://en.wikipedia.org/wiki/Convolutional_neural_networks) have been applied to fields including [computer](https://en.wikipedia.org/wiki/Computer_vision) [vision](https://en.wikipedia.org/wiki/Computer_vision), [speech recognition](https://en.wikipedia.org/wiki/Automatic_speech_recognition), [natural language processing,](https://en.wikipedia.org/wiki/Natural_language_processing) audio recognition, social network filtering, [machine translation](https://en.wikipedia.org/wiki/Machine_translation), [bioinformatics](https://en.wikipedia.org/wiki/Bioinformatics), [drug design,](https://en.wikipedia.org/wiki/Drug_design) medical image analysis, material inspection and [board game](https://en.wikipedia.org/wiki/Board_game) programs, where they have produced results comparable to and in some cases superior to human experts.

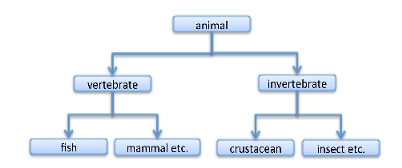


### 1 pMdN4mztA49_n0YSP7EpIQ

**Fig. 3.2 Perceptron**

### 3.4 Kernel in CNN

Before we can jump into kernels, we need to understand what a support vector machine is. Support Vector Machine or SVM are supervised learning models with associated learning algorithms that analyse data for classification (classification means knowing what belong to what e.g. ‘apple’ belongs to class ‘fruit’ while ‘dog’ to class ‘animals’ -see fig)

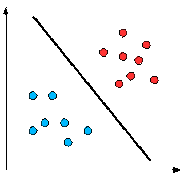


**Fig. 3.3 Classification**

In support vector machines, it looks somewhat like figure below which separates the blue balls from red.

SVM is a classifier formally defined by a separating hyperplane. A hyperplane is a subspace of one dimension less than its ambient space. The dimension of a mathematical space (or object) is informally defined as the minimum number of coordinates (x,y,z axis) needed to specify any point (like each blue and red point) within it while an ambient space is a space surrounding a mathematical object. A mathematical object is an abstract object arising in mathematics An abstract object is an object which does not exist at any particular time or place, but rather exists as a type of thing, i.e., an idea, or abstraction

Therefore the hyperplane of a two-dimensional space below is a one-dimensional line dividing the red and blue dots.

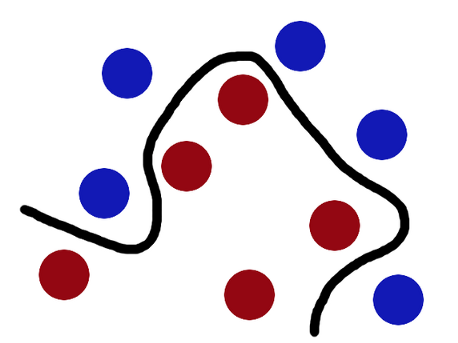


From the example above of trying to predict the breed of a particular dog, it goes like this

Data (all breeds of dog)→ Features(skin colour, hair, etc)→ Learning algorithm

### 3.5 Why Kernels

In the figure below



We can’t solve the above problem linearly like we did before. The red and blue balls cannot be separated by a straight line as they are randomly distributed and this, in reality, is how most real life problem data are -randomly distributed.

In machine learning, a “kernel” is usually used to refer to the kernel trick, a method of using a linear classifier to solve a non-linear problem. It entails transforming linearly inseparable data into linearly separable ones. The kernel function is what is applied on each data instance to map the original non-linear observations into a higher-dimensional space in which they become separable.

Using the dog breed prediction example again, kernels offer a better alternative. Instead of defining a slew of features, you define a single kernel function to compute similarity between breeds of dog. You provide this kernel, together with the data and labels to the learning algorithm, and out comes a classifier.

### 3.6 Working of Kernel

To better understand how Kernels work, let us use Lili Jiang’s mathematical [illustration](https://www.quora.com/What-are-Kernels-in-Machine-Learning-and-SVM/answer/Lili-Jiang?srid=oOgT)

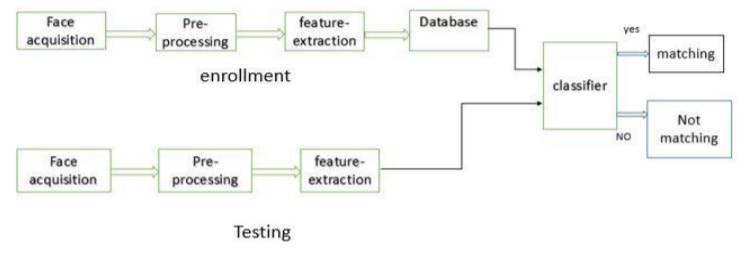
**Mathematical definition**:

K(x, y) = <f(x), f(y)>. …(1)

Here K is the kernel function, x, y are n dimensional inputs. f is a map from n-dimension to m-dimension space. < x,y> denotes the dot product. Usually m is much larger than n.

**Intuition**: normally calculating <f(x), f(y)> requires us to calculate f(x), f(y) first, and then do the dot product. These two computation steps can be quite expensive as they involve manipulations in m dimensional space, where m can be a large number. But after all the trouble of going to the high dimensional space, the result of the dot product is really a scalar: we come back to one-dimensional space again. Now, the question we have is: do we really need to go through all the trouble to get this one number? Do we really have to go to the m-dimensional space? The answer is no, if you find a clever kernel.

### 3.7 Proposed System



The block diagram show the simple explanation of building database and testing system.

1. Face acquisition phase: In the initial phase, the image is captured by means of any source such as camera or working with standard databases of images. The captured image is converted to grey scale image and resized to eliminate pose problem. The pre-processing is done to remove noises from images. Further Different simple methods are carried to remove illumination and darkness or brightness problems.
2. Feature Detection phase: In this phase, facial features are extracted using different technique like Edge Detection Techniques using Haar Cascade detection which in turn is powered by integral images and Adaboost technique. All this methods are useful which gives desire features vectors of selected region.
3. Classification phase: During this phase, after recognition of object in the form of features vectors classification is done using different distance methods like Haar cascade classification and then use them for training the images algorithm of frontal face named as haarcascade\_frontalface\_default.xml followed by
4. Recognition phase: testing the new images or the test images of databases for recognition of correct image by using Local Binary Patterns Histogram (LBPH) algorithm. Since it is based on local binary operator. It is widely used in facial recognition due to its computational simplicity and discriminative power.

### 3.8 Haar Cascade

Haar Cascade is a machine learning object detection algorithm used to identify objects in an image or video and based on the concept of ​​ features proposed by Paul Viola and Michael Jones in their paper "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001.  
  
It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

The detection of face is a process carried out using Haar cascade classifiers due to its speed. Haar Classifier is a supervised classifier and can be trained to detect faces in an image. It may be seen as a funnel where each region of any given image is processed using a set of classifiers termed as Haar-features. They behave as a funnel called the Haar Cascade. Classifiers at the top of the cascade are very fast and their false negative rate is very low. They discard regions of an image that does not contains any face. The features become more complex further down the cascade and images are rejected immediately if the features do not resemble a face. The integral of a grayscale image is calculated by the e right of the corresponding pixel. Thus, calculation of average intensity of any rectangular portion of an image may be calculated with the help of only 4 pixels at a time. The mathematical representation of haar classifier is given in equation.

Sum =I(C) +I(A)-I(B)-I(D) …(2)

The algorithm has four stages:

1. Haar Feature Selection
2. Creating  Integral Images
3. Adaboost Training
4. Cascading Classifiers

It is well known for being able to detect faces and body parts in an image, but can be trained to identify almost any object.

Let’s take face detection as an example. Initially, the algorithm needs a lot of positive images of faces and negative images without faces to train the classifier.   Then we need to extract features from it.

First step is to collect the Haar Features.  A Haar feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums.

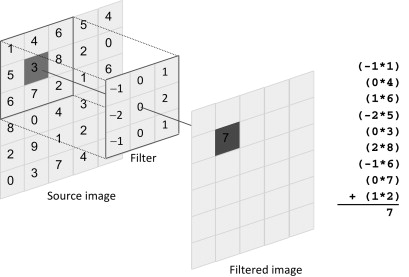
### 3.8.1 Viola face detection algorithm

1. Haar features:

Haar features are similar to these convolution kernels which are used to detect the presence of that feature in the given image.

Detect edges using convolution kernels:

Given an input image and convolution kernel, we place kernel to a corner and do convolution multiplication shifting the kernels.

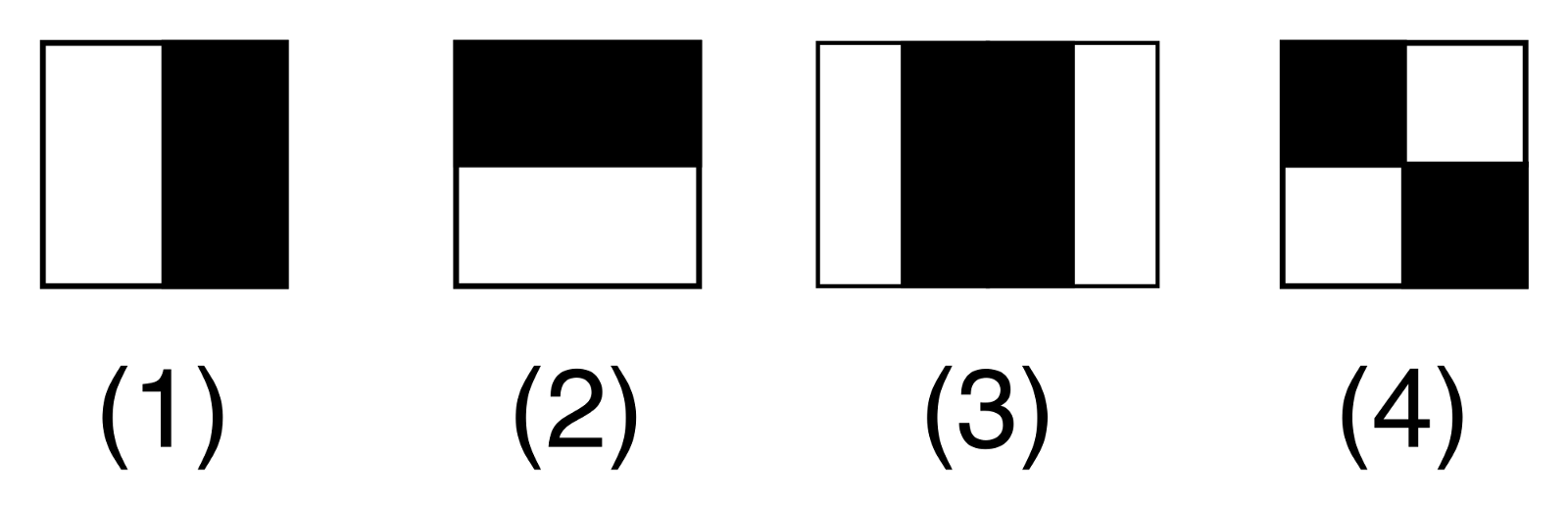


**Fig 3.7: convolution multiplication shifting the kernels.**

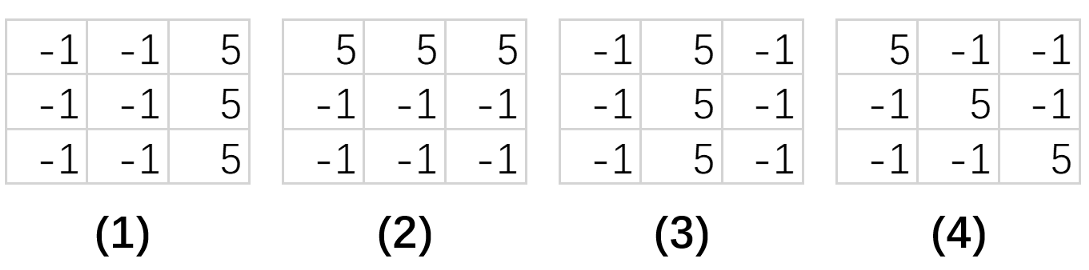
This method is used to detect different types of edges using different kernels.

A Haar-Feature is just like a kernel in CNN, except that in a CNN, the values of the kernel are determined by training, while a Haar-Feature is manually determined.

Here are some Haar-Features. The first two are “edge features”, used to detect edges. The third is a “line feature”, while the fourth is a “four rectangle feature”, most likely used to detected a slanted line.



Numerically, they might look something like this:



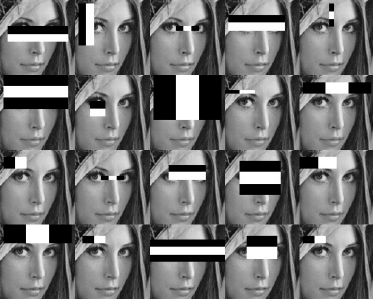
**Fig 3.9: Haar-features represented numerically**

As explained, each the 3x3 kernel moves across the image and does matrix multiplication with every 3x3 part of the image, emphasizing some features and smoothing others.

Haar-Features are good at detecting edges and lines. This makes it especial effective in face detection.

When haar features are applied to image to a girl.

Each feature results in a single value which is calculated by subtracting the sum of pixels under a white rectangle from the sum of pixels under the black rectangle.



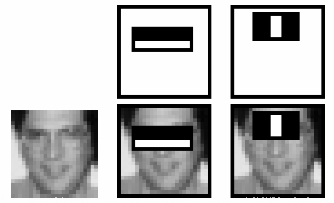
Every haar feature has some sort of resemblance to identify a part of the face. Viola-Jones uses 24\*24 as base window size and calculates the above features all over the image shifting by 1 PX.

If we consider all possible parameters of the haar features like position, scale, and type we end up calculating about 160,000+ features. So we need to evaluate a huge set of features for every 24\*24 PX.

So to avoid this we have an idea to avoid redundant features and pick only those features which are very useful for us. This can be done using AdaBoost.

1. Integral Images

It is used to make this process super-fast. But among all these features we calculated, most of them are irrelevant. For example, consider the image below. Top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose.  But the same windows applying on cheeks or any other place is irrelevant.



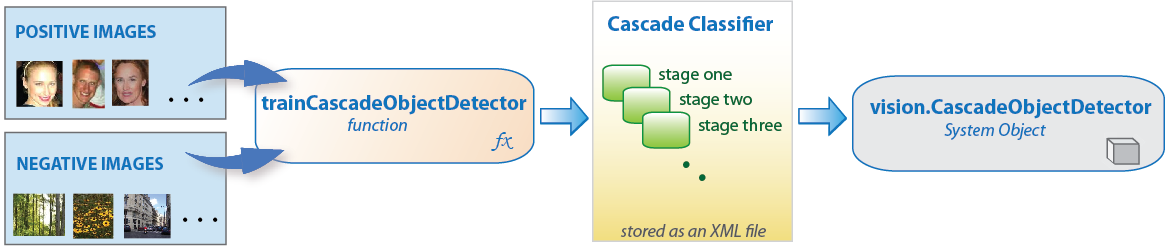
Then that is why we select the best features out of 160000+ using Adaboost

1. Adaboost

This is accomplished using a concept called Adaboost which both selects the best features and trains the classifiers that use them. This algorithm constructs a “strong” classifier as a linear combination of weighted simple “weak” classifiers.  The process is as follows.

During the detection phase, a window of the target size is moved over the input image, and for each subsection of the image and Haar features are calculated.   You can see this in action in the video below.  This difference is then compared to a learned threshold that separates non-objects from objects.  Because each Haar feature is only a "weak classifier" (its detection quality is slightly better than random guessing) a large number of Haar features are necessary to describe an object with sufficient accuracy and are therefore organized into cascade classifiers to form a strong classifier.

1. Cascade Classifier



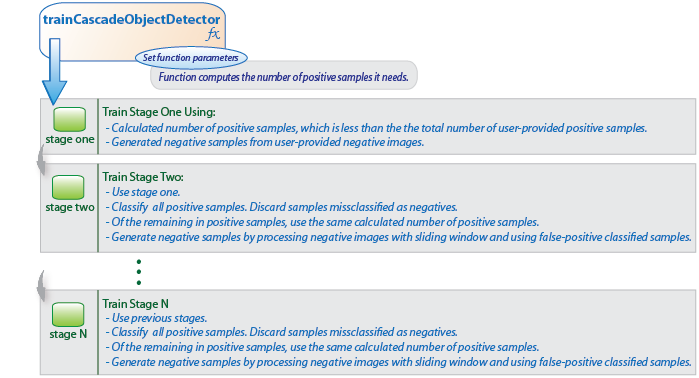
The cascade classifier consists of a collection of stages, where each stage is an ensemble of weak learners. The weak learners are simple classifiers called *decision stumps*. Each stage is trained using a technique called boosting. *Boosting* provides the ability to train a highly accurate classifier by taking a weighted average of the decisions made by the weak learners.

Each stage of the classifier labels the region defined by the current location of the sliding window as either positive or negative. *Positive* indicates that an object was found and *negative* indicates no objects were found. If the label is negative, the classification of this region is complete, and the detector slides the window to the next location. If the label is positive, the classifier passes the region to the next stage. The detector reports an object found at the current window location when the final stage classifies the region as positive.

The stages are designed to reject negative samples as fast as possible. The assumption is that the vast majority of windows do not contain the object of interest. Conversely, true positives are rare and worth taking the time to verify.

* A *true positive* occurs when a positive sample is correctly classified.
* A *false positive* occurs when a negative sample is mistakenly classified as positive.
* A *false negative* occurs when a positive sample is mistakenly classified as negative.

To work well, each stage in the cascade must have a low false negative rate. If a stage incorrectly labels an object as negative, the classification stops, and you cannot correct the mistake. However, each stage can have a high false positive rate. Even if the detector incorrectly labels a non-object as positive, you can correct the mistake in subsequent stages.  Adding more stages reduces the overall false positive rate, but it also reduces the overall true positive rate.



**Fig 3.13: Training the Cascade Classifier**

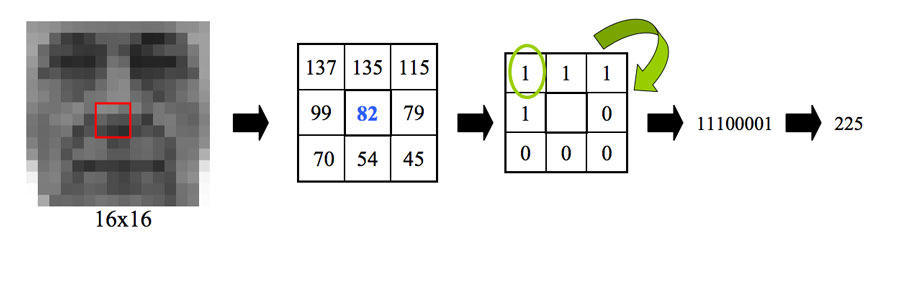
Cascade classifier training requires a set of positive samples and a set of negative images. You must provide a set of positive images with regions of interest specified to be used as positive samples. You can use the Image Labeler to label objects of interest with bounding boxes. The Image Labeler outputs a table to use for positive samples. You also must provide a set of negative images from which the function generates negative samples automatically. To achieve acceptable detector accuracy, set the number of stages, feature type, and other function parameters.

### 3.9 Local binary patterns histograms (LBPH) Face Recognizer

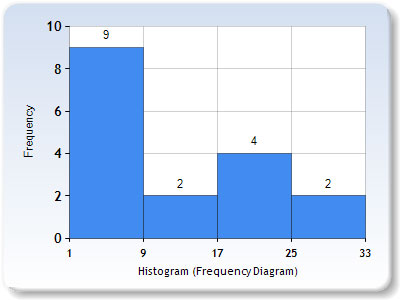
We know that Eigen faces and Fisher faces are both affected by light and, in real life, we can't guarantee perfect light conditions. LBPH face recognizer is an improvement to overcome this drawback. The idea with LBPH is not to look at the image as a whole, but instead, try to find its local structure by comparing each pixel to the neighbouring pixels.

It takes a 3×3 window and move it across one image. At each move (each local part of the picture), compare the pixel at the centre, with its surrounding pixels. Denote the neighbours with intensity value less than or equal to the centre pixel by 1 and the rest by 0.

After we read these 0/1 values under the 3×3 window in a clockwise order, we will have a binary pattern like 11100011 that is local to a particular area of the picture. When we finish doing this on the whole image, we will have a list of local binary patterns.



We get a list of local binary patterns, you convert each one into a decimal number using [binary to decimal conversion](https://www.mathsisfun.com/binary-number-system.html) (as shown in above image) and then you make a [histogram](https://www.mathsisfun.com/data/histograms.html) of all of those decimal values. A sample histogram looks like this:



In the end, we will have one histogram for each face in the training data set. That means that if there were 100 images in the training data set then LBPH will extract 100 histograms after training and store them for later recognition. Remember, the algorithm also keeps track of which histogram belongs to which person.

Later during recognition, the process is as follows:

1. Feed a new image to the recognizer for face recognition.
2. The recognizer generates a histogram for that new picture.
3. It then compares that histogram with the histograms it already has.
4. Finally, it finds the best match and returns the person label associated with that best match.

We can perform and see that the LBP faces are not affected by changes in light conditions.

### 3.10 Database

SQLite is a self-contained, high-reliability, embedded, full-featured, public-domain, SQL database engine. It is the most used database engine in the world. It is an in-process library and its code is publicly available. It is free for use for any purpose, commercial or private. It is basically an embedded SQL database engine. Ordinary disk files can be easily read and write by SQLite because it does not have any separate server like SQL. The SQLite database file format is cross-platform so that anyone can easily copy a database between 32-bit and 64-bit systems. Due to all these features, it is a popular choice as an Application File Format.

### 3.10.1 Features of SQLite

1. The transactions follow ACID properties i.e. atomicity, consistency, isolation, and durability even after system crashes and power failures.
2. The configuration process is very easy, no setup or administration needed.
3. All the features of SQL are implemented in it with some additional features like partial indexes, indexes on expressions, JSON, and common table expressions.
4. Sometimes it is faster than the direct file system I/O.
5. It supports terabyte-sized databases and gigabyte-sized strings and blobs.
6. Almost all OS supports SQLite like Android, BSD, iOS, Linux, Mac, Solaris, VxWorks, and Windows (Win32, WinCE, etc. It is very much easy to port to other systems.
7. Complete database can be stored in a single cross-platform disk file.

### 3.10.2 Applications of SQLite

1. Due to its small code print and efficient usage of memory, it is the popular choice for the database engine in cell phones, PDAs, MP3 players, set-top boxes, and other electronic gadgets.
2. It is used as an alternative for open to writing XML, JSON, CSV or some proprietary format into disk files used by the application.
3. As it has no complication for configuration and easily stores file in an ordinary disk file, so it can be used as a database for small to medium sized websites.
4. It is faster and accessible through a wide variety of third-party tools, so it has great application in different software platforms.

To use the SQLite3 module we need to add an import statement to our python script:

We use the function sqlite3.connect to connect to the database. When we are done working with the DB we need to close the connection by committing and we can use sqlite3.commit and sqlite3.close

### 3.11 SMTP Protocol

SMTP is part of the application layer of the TCP/IP protocol. Using a process called "store and forward," SMTP moves your email on and across networks. It works closely with something called the Mail Transfer Agent (MTA) to send your communication to the right computer and email inbox.

### 3.11.1 SMTP at work.

SMTP provides a set of codes that simplify the communication of email messages between email servers (the network computer that handles email coming to you and going out). It's a kind of shorthand that allows a server to break up different parts of a message into categories the other server can understand. When you send a message out, it's turned into strings of text that are separated by the code words (or numbers) that identify the purpose of each section.

SMTP provides those codes, and email server software is designed to understand what they mean. As each message travels towards its destination, it sometimes passes through a number of computers as well as their individual MTAs. As it does, it's briefly stored before it moves on to the next computer in the path. Think of it as a letter going through different hands as it winds its way to the right mailbox.

class smtplib.SMTP\_SSL(host='', port=0, local\_hostname=None, keyfile=None, certfile=None, [timeout, ]context=None, source\_address=None)

An [SMTP\_SSL](https://docs.python.org/3/library/smtplib.html#smtplib.SMTP_SSL) instance behaves exactly the same as instances of [SMTP](https://docs.python.org/3/library/smtplib.html#smtplib.SMTP). [SMTP\_SSL](https://docs.python.org/3/library/smtplib.html#smtplib.SMTP_SSL) should be used for situations where SSL is required from the beginning of the connection and using starttls() is not appropriate. If host is not specified, the local host is used. If port is zero, the standard SMTP-over-SSL port (465) is used. The optional arguments local\_hostname, timeout and source\_address have the same meaning as they do in the [SMTP](https://docs.python.org/3/library/smtplib.html#smtplib.SMTP) class. context, also optional, can contain a [SSLContext](https://docs.python.org/3/library/ssl.html#ssl.SSLContext) and allows configuring various aspects of the secure connection.

### 3.12 Email Protocol to Send Alert

The [email](https://docs.python.org/3/library/email.html#module-email) package is a library for managing email messages. It is specifically not designed to do any sending of email messages to SMTP ([RFC 2821](https://tools.ietf.org/html/rfc2821.html)), NNTP, or other servers; those are functions of modules such as [smtplib](https://docs.python.org/3/library/smtplib.html#module-smtplib) and [nntplib](https://docs.python.org/3/library/nntplib.html#module-nntplib). The [email](https://docs.python.org/3/library/email.html#module-email) package attempts to be as RFC-compliant as possible, supporting [RFC 5233](https://tools.ietf.org/html/rfc5233.html) and [RFC 6532](https://tools.ietf.org/html/rfc6532.html), as well as such MIME-related RFCs as [RFC 2045](https://tools.ietf.org/html/rfc2045.html), [RFC 2046](https://tools.ietf.org/html/rfc2046.html), [RFC 2047](https://tools.ietf.org/html/rfc2047.html), [RFC 2183](https://tools.ietf.org/html/rfc2183.html), and [RFC 2231](https://tools.ietf.org/html/rfc2231.html).

The overall structure of the email package can be divided into three major components, plus a fourth component that controls the behavior of the other components.

The central component of the package is an “object model” that represents email messages. An application interacts with the package primarily through the object model interface defined in the [message](https://docs.python.org/3/library/email.message.html#module-email.message) sub-module. The application can use this API to ask questions about an existing email, to construct a new email, or to add or remove email subcomponents that themselves use the same object model interface. That is, following the nature of email messages and their MIME subcomponents, the email object model is a tree structure of objects that all provide the [EmailMessage](https://docs.python.org/3/library/email.message.html#email.message.EmailMessage) API.

The other two major components of the package are the [parser](https://docs.python.org/3/library/email.parser.html#module-email.parser) and the [generator](https://docs.python.org/3/library/email.generator.html#module-email.generator). The parser takes the serialized version of an email message (a stream of bytes) and converts it into a tree of [EmailMessage](https://docs.python.org/3/library/email.message.html#email.message.EmailMessage) objects. The generator takes an [EmailMessage](https://docs.python.org/3/library/email.message.html#email.message.EmailMessage) and turns it back into a serialized byte stream. (The parser and generator also handle streams of text characters, but this usage is discouraged as it is too easy to end up with messages that are not valid in one way or another.)

The control component is the [policy](https://docs.python.org/3/library/email.policy.html#module-email.policy) module. Every [EmailMessage](https://docs.python.org/3/library/email.message.html#email.message.EmailMessage), every [generator](https://docs.python.org/3/library/email.generator.html#module-email.generator), and every [parser](https://docs.python.org/3/library/email.parser.html#module-email.parser) has an associated [policy](https://docs.python.org/3/library/email.policy.html#module-email.policy) object that controls its behavior. Usually an application only needs to specify the policy when an [EmailMessage](https://docs.python.org/3/library/email.message.html#email.message.EmailMessage) is created, either by directly instantiating an [EmailMessage](https://docs.python.org/3/library/email.message.html#email.message.EmailMessage) to create a new email, or by parsing an input stream using a [parser](https://docs.python.org/3/library/email.parser.html#module-email.parser). But the policy can be changed when the message is serialized using a [generator](https://docs.python.org/3/library/email.generator.html#module-email.generator). This allows, for example, a generic email message to be parsed from disk, but to serialize it using standard SMTP settings when sending it to an email server.

The email package does its best to hide the details of the various governing RFCs from the application. Conceptually the application should be able to treat the email message as a structured tree of Unicode text and binary attachments, without having to worry about how these are represented when serialized. In practice, however, it is often necessary to be aware of at least some of the rules governing MIME messages and their structure, specifically the names and nature of the MIME “content types” and how they identify multipart documents. For the most part this knowledge should only be required for more complex applications, and even then it should only be the high level structure in question, and not the details of how those structures are represented. Since MIME content types are used widely in modern internet software (not just email), this will be a familiar concept to many programmers.

### 3.13 Face Detection Tools

Face detection tools are used as a hardware in this project to perform it. Face detection detects the student when students passing from there. Face detection tools are as follows:

### 3.13.1 Camera

A camera is an optical instrument to capture still images or to record memory image, which are stored in a physical medium such as in a digital system or on [p](https://en.wikipedia.org/wiki/Photographic_film)hotographic film. A camera consists of a lens which focuses light from the scene, and a camera body which holds the image capture mechanism.

### 3.13.2 CCTV Surveillance

Closed circuit television (CCTV) is also known as video surveillance. It is a system where all the elements like video camera, display monitors, recording devices are directly connected.

### 3.14 Language and Library

Programming language is used for the coding. In this project python 3.6 is used. Python 3 makes coding more obvious and intuitive by removing duplicate constructs and modules; Python 3 simplifies multilingual support, with its core string type based on Unicode by default; Python 3 makes it easier to swap in any print function, now that Print () is built-in (rather than a statement)

### 3.14.1 Python 3.6

Python is a[n open source](https://simple.wikipedia.org/wiki/Open_source) [programming language](https://simple.wikipedia.org/wiki/Programming_language) that was made to be easy-to-read and powerful. A [Dutch](https://simple.wikipedia.org/wiki/Netherlands) programmer named [Guido van Rossum](https://simple.wikipedia.org/wiki/Guido_van_Rossum) made Python in 1991. He named it after the television show [Monty Python's Flying Circus](https://simple.wikipedia.org/wiki/Monty_Python%27s_Flying_Circus). Many Python examples and tutorials include jokes from the show. Python is an interpreted language. Interpreted languages do not need to be [compiled](https://simple.wikipedia.org/wiki/Compiled_language) to run. A program called an [interpreter](https://simple.wikipedia.org/wiki/Interpreter_(computing)) runs Python code on almost any kind of computer. This means that a programmer can change the code and quickly see the results. This also means Python is slower than a compiled language like C, because it is not running [machine](https://simple.wikipedia.org/wiki/Machine_code) [code](https://simple.wikipedia.org/wiki/Machine_code) directly.

### 3.14.2 Coding Face Recognition using Python and OpenCV

### We are going to divide the Face Recognition process in this tutorial into three steps:

1. **Prepare Training Data:** Read training images for each person/subject along with their labels, detect faces from each image and assign each detected face an integer label of the person it belongs.
2. **Train Face Recognizer:** Train OpenCV's LBPH recognizer by feeding it the data we prepared in step 1.
3. **Prediction:** Introduce some test images to face recognizer and see if it predicts them correctly.

Before we start the actual coding, we need to install the Code Dependencies and import the Required Modules:

**3.14.2.1 Code Dependencies**

Install the following dependencies:

1. [OpenCV 3.2.0](http://opencv.org/releases.html)
2. Python v3.6
3. [NumPy](http://www.numpy.org/) that makes computing in Python easy. It contains a powerful implementation of N-dimensional arrays which we will use for feeding data as input to OpenCV functions.
4. Dlib: According to dlib’s [github page](http://github.com/davisking/dlib), dlib is *a toolkit for making real world machine learning and data analysis applications in C++.* While the library is originally written in C++, it has good, easy to use Python bindings. The frontal face detector in dlib works really well. It is simple and just works out of the box.
5. Pillow: Pillow is a Python Imaging Library (PIL), which adds support for opening, manipulating, and saving images. The current version identifies and reads a large number of formats.

**3.14.2.2 Required Modules**

Import the following modules:

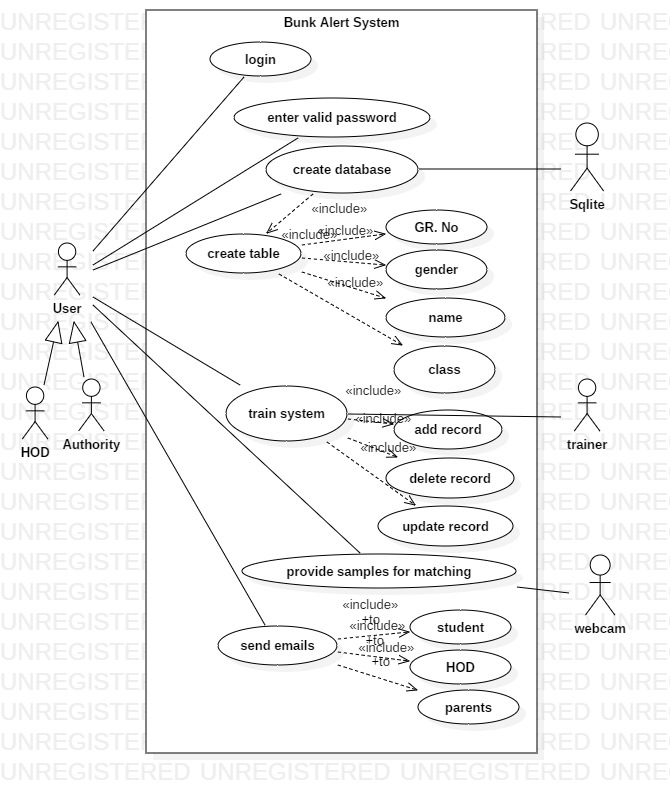
* **cv2:** This is the OpenCV module for Python used for face detection and face recognition.
* **os:** We will use this Python module to read our training directories and file names.
* **numpy:** This module converts Python lists to numpy arrays as OpenCV face recognizer needs them for the face recognition process.
* **Sqlite3:** SQLite3 is a very easy to use database engine. It is self-contained, serverless, zero-configuration and transactional. The Python Standard Library includes a module called "sqlite3" intended for working with this database. This module is a SQL interface compliant with the DB-API 2.0 specification.
* **PyQt5:** 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++
* **Subprocess:** The subprocess module present in Python(both 2. x and 3. x) is used to run new applications or programs through Python code by creating new processes. It also helps to obtain the input/output/error pipes as well as the exit codes of various commands.
* **Sys:** System-specific parameters and functions. This module provides access to some variables used or maintained by the interpreter and to functions that interact strongly with the interpreter. It is always available.
* **Email:** You can create a new object structure by creating [Message](https://docs.python.org/2/library/email.message.html#email.message.Message) instances, adding attachments and all the appropriate headers manually. For MIME messages though, the [email](https://docs.python.org/2/library/email.html#module-email) package provides some convenient subclasses to make things easier.
* **Smtplib:** The smtplib module defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon. For details of SMTP and ESMTP operation, consult RFC 821 (Simple Mail Transfer Protocol) and RFC 1869 (SMTP Service Extensions).

### 3.15 UML Diagram

Use case diagrams are usually referred to as [behavior diagrams](https://www.uml-diagrams.org/uml-25-diagrams.html#behavior-diagram) used to describe a set of actions ([use cases](https://www.uml-diagrams.org/use-case.html)) that some system or systems ([subject](https://www.uml-diagrams.org/use-case-subject.html)) should or can perform in collaboration with one or more external users of the system ([actors](https://www.uml-diagrams.org/use-case-actor.html)). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

Note, that UML 2.0 to 2.4 specifications also described use case diagram as a specialization of a [class diagram](https://www.uml-diagrams.org/class-diagrams-overview.html), and class diagram is a [structure diagram](https://www.uml-diagrams.org/uml-25-diagrams.html#structure-diagram).

Use case diagrams are in fact twofold - they are both [behavior diagrams](https://www.uml-diagrams.org/uml-25-diagrams.html#behavior-diagram), because they describe behavior of the system, and they are also [structure diagrams](https://www.uml-diagrams.org/uml-25-diagrams.html#structure-diagram) - as a special case of class diagrams where classifiers are restricted to be either [actors](https://www.uml-diagrams.org/use-case-actor.html) or [use cases](https://www.uml-diagrams.org/use-case.html) related to each other with [associations](https://www.uml-diagrams.org/association.html).



### 3.15.1 Class Diagram:

The class diagram is the main building block of [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) modeling. It is used for general [conceptual modeling](https://en.wikipedia.org/wiki/Conceptual_model) of the structure of the application, and for detailed modeling translating the models into [programming code](https://en.wikipedia.org/wiki/Programming_code). Class diagrams can also be used for [data modeling](https://en.wikipedia.org/wiki/Data_modeling). The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

In the diagram, classes are represented with boxes that contain three compartments:

The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.

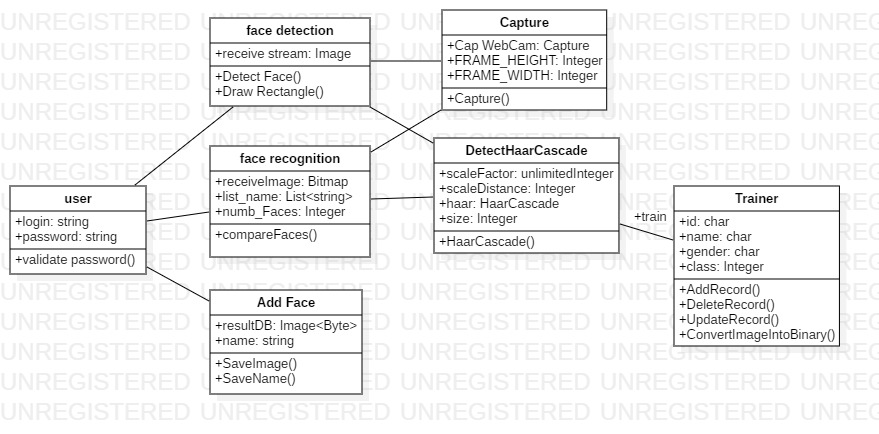
The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.

The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

A class with three compartments.

In the design of a system, a number of classes are identified and grouped together in a class diagram that helps to determine the static relations between them. With detailed modeling, the classes of the conceptual design are often split into a number of subclasses.

In order to further describe the behaviour of systems, these class diagrams can be complemented by a [state diagram](https://en.wikipedia.org/wiki/State_diagram) or [UML state machine](https://en.wikipedia.org/wiki/UML_state_machine).



### 3.15.2 Activity Diagram:

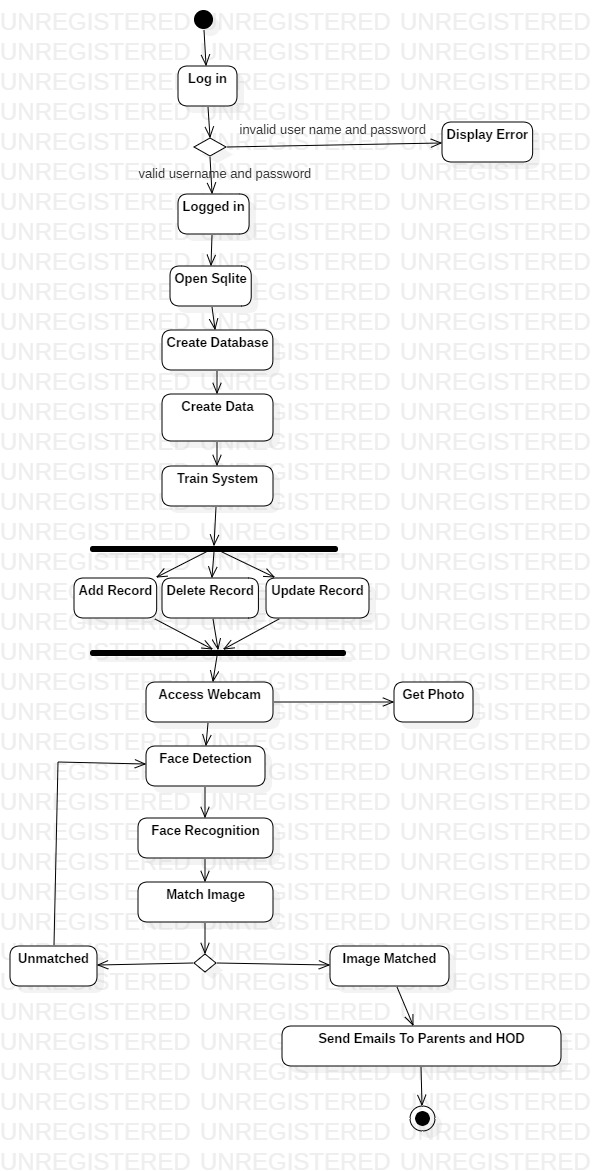
Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

The purpose of an activity diagram can be described as −

* Draw the activity flow of a system.
* Describe the sequence from one activity to another.
* Describe the parallel, branched and concurrent flow of the system.

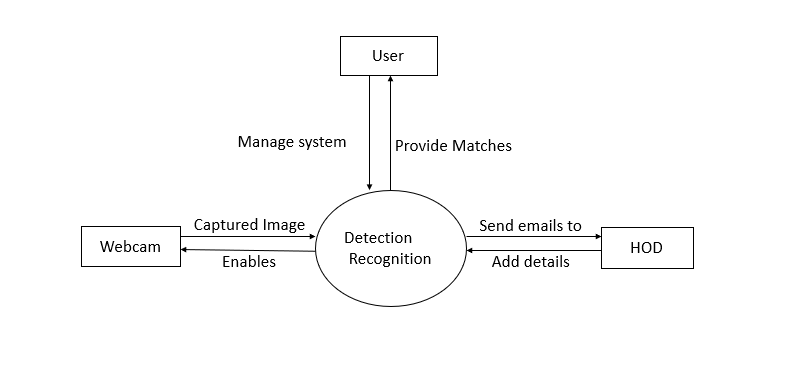


### 3.15.3 Dataflow diagram:

In Software engineering DFD (data flow diagram) can be drawn to represent the system of different levels of abstraction. Higher level DFDs are partitioned into low levels-hacking more information and functional elements. Levels in DFD are numbered 0, 1, 2 or beyond. Here, we will see mainly 3 levels in data flow diagram, which are: 0-level DFD, 1-level DFD, and 2-level DFD.

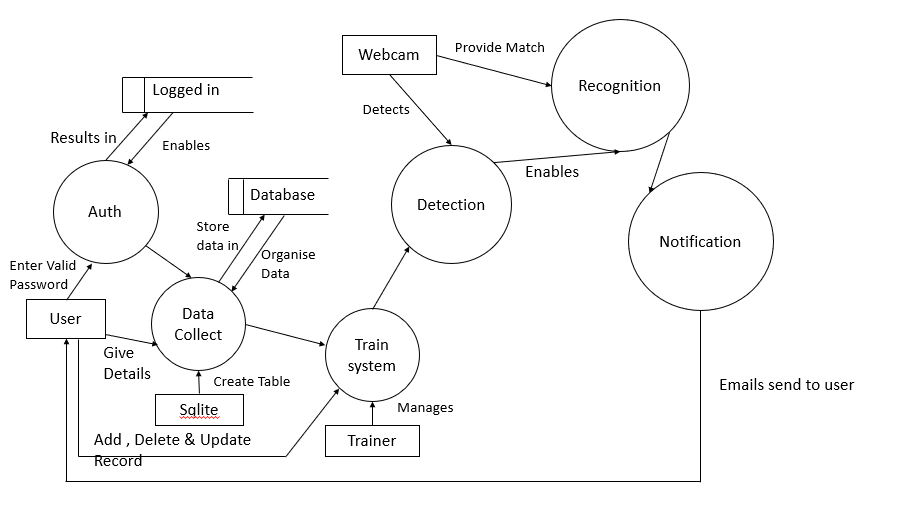
1. Level 0:

It is also known as context diagram. It’s designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represent the entire system as single bubble with input and output data indicated by incoming/outgoing arrows.



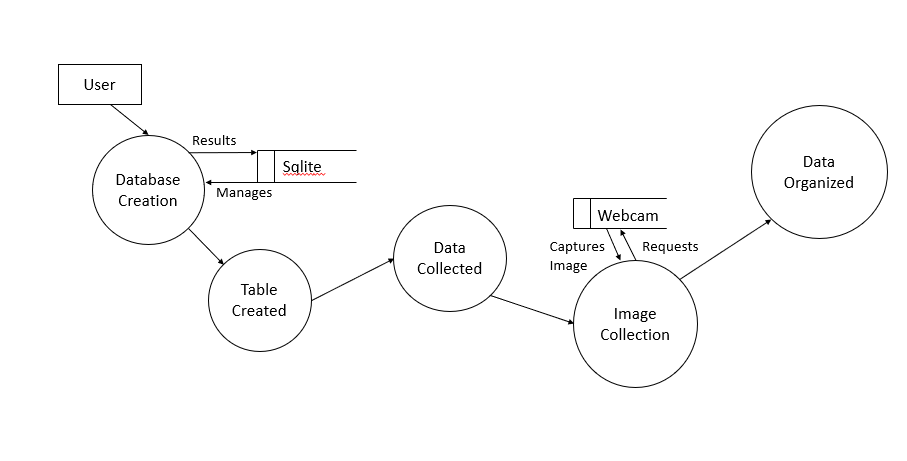
1. Level 1:

In 1-level DFD, context diagram is decomposed into multiple bubbles/processes.in this level we highlight the main functions of the system and breakdown the high level process of 0-level DFD into sub-processes.



1. Level 2:

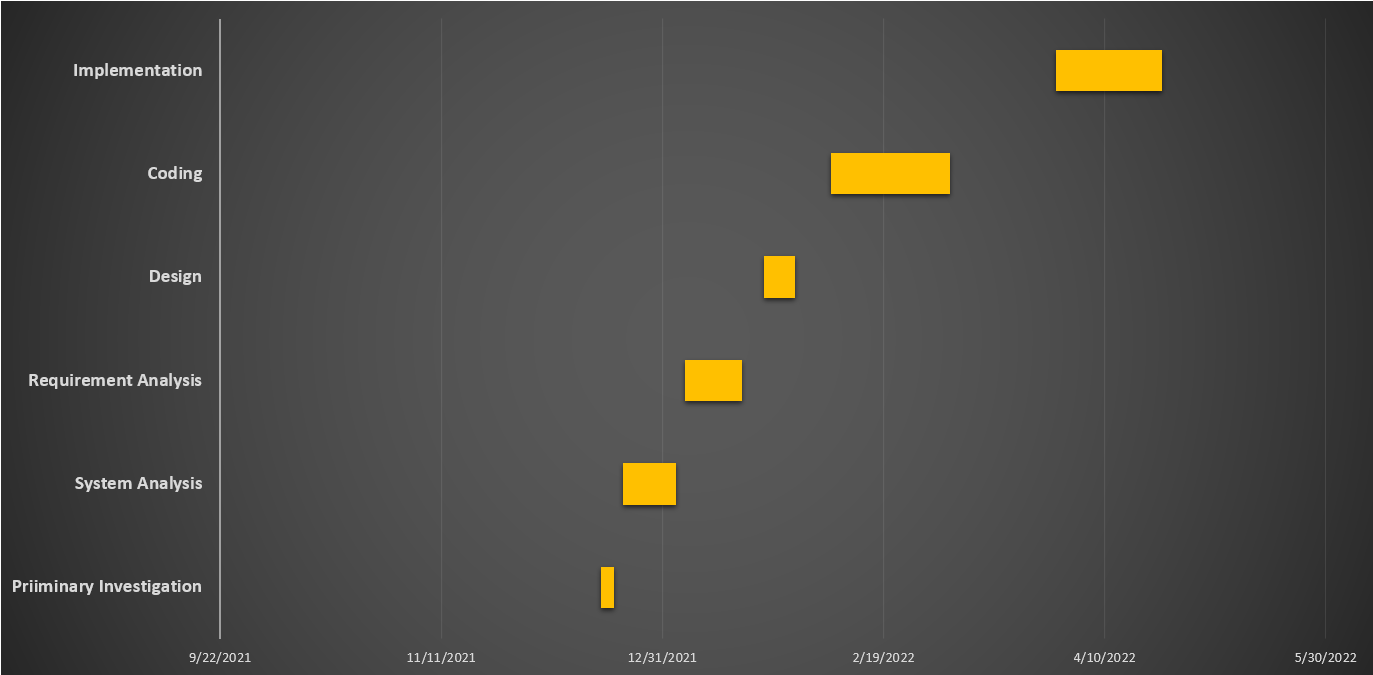
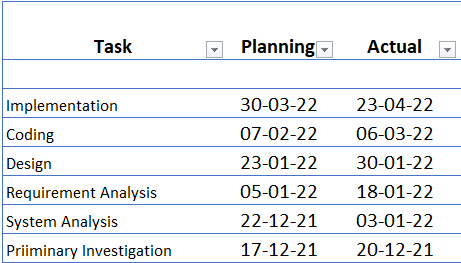
2-level DFD goes one step deeper into parts of 1-level DFD. It can be used to plan or record the specific/necessary detail about the system’s functioning.



### 3.15.4 ER Diagram

# 

### 3.15.6 Gantt Chart



# 3.15.7 Flow Chart:

A **flowchart** is a type of diagram that represents a workflow or process. Flowcharts are used in analysing, designing, documenting or managing a process or program in various fields. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.

The flowchart below shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problems.

# 

# CHAPTER 4

## Experimental Procedure and Result

In this section, we report the experimental procedures and results obtained for haar cascade which detect the frontal face as well as classify the extracted features.

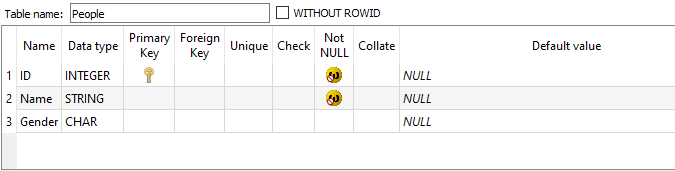
### 4.1 Dataset generation:

### 4.1.1 Sqlite3

SQLite is a popular choice asembedded database software for local/client storage inapplication software. It is arguably the most widely deployeddatabase engine, as it is used today by several widespread browsers,operating systems, andembedded systems, among others. SQLite hasbindings to many programming languages.

Two tables are made within a database named Dataset which are as follows

People:

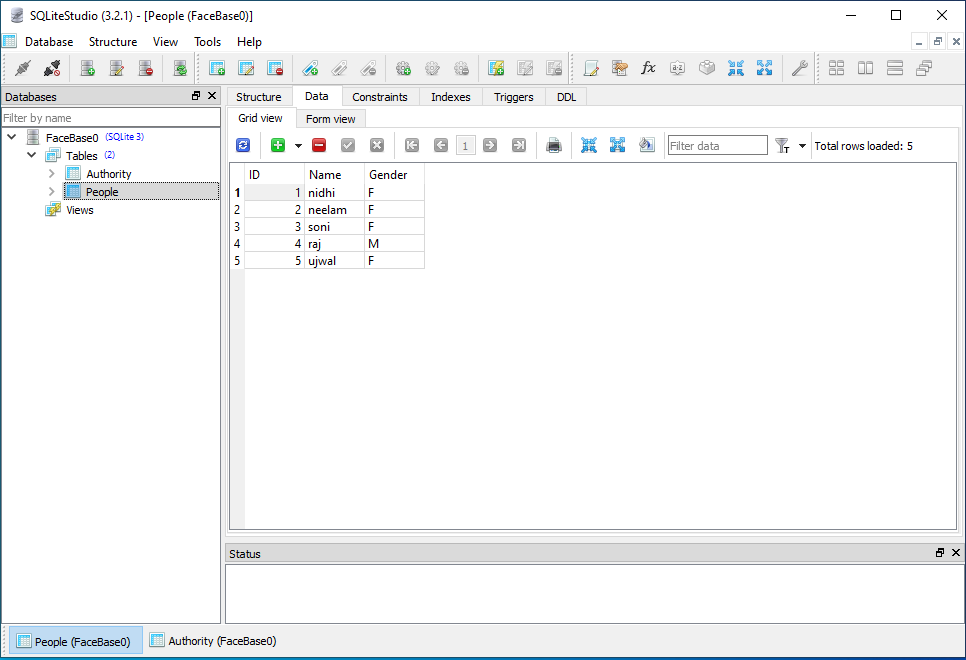


**Students Table Named as People**

Which contains details about students in academic year with their ID, Name, and Gender

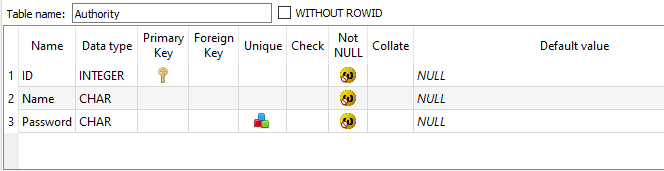
This table must be filled before taking the photos of students to store in the database since while storing the photos to the dataset folder the application will ask for students ID.

Entered details looks as follows,



Details can extend as per required.

Authority:



Which contains details of authority who can login into system using ID, Name and Password

If the details are wrong an alert window will pop with caption of “Enter correct details”.

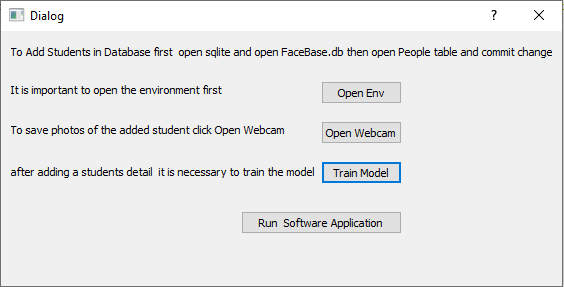
Commands used are as follows

sqlite3.connect(database [,timeout ,other optional arguments])

connection.cursor([cursorClass])

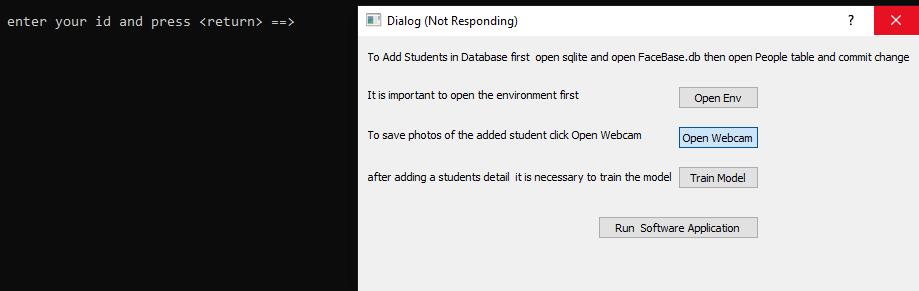
cursor.execute(sql [, optional parameters])

connection.commit() connection.close()



It is necessary to open the environment myenv in which this system works since all the needed packages are in myenv for this application to function for we have given an open env button which makes it easy for any user to open the environment.

When the Open Webcam button is clicked the system will ask for student ID and on next input Name which we have to put in double quotes since behind the scenes sql data is been stored as name/string, so for the system can connect the ID to name of student and store it in faceBase folder.



Then webcams open up to store 50 photos of the student and the student must move his/her face slightly so that the system can take photos from various poses. Here cv2 and haar cascade classifier comes in to work

### 4.1.2 Opencv2, Haar cascade classifier and LBPH recognizer

The haarcascade\_frontalface\_default.xml is a haar cascade designed by OpenCV to detect the frontal face. A Haar Cascade works by training the cascade on thousands of negative images with the positive image superimposed on it. The haar cascade is capable of detecting features from the source. We also use detectMultiScale. Function, reduce the width and height of your image. The smaller your image is, the less data there is to process, and thus the detector will run faster.

All the detected photos of a person are stored in the dataset folder with naming  of for eg:\*.1 till \*.50 and for each person it is done in the same way

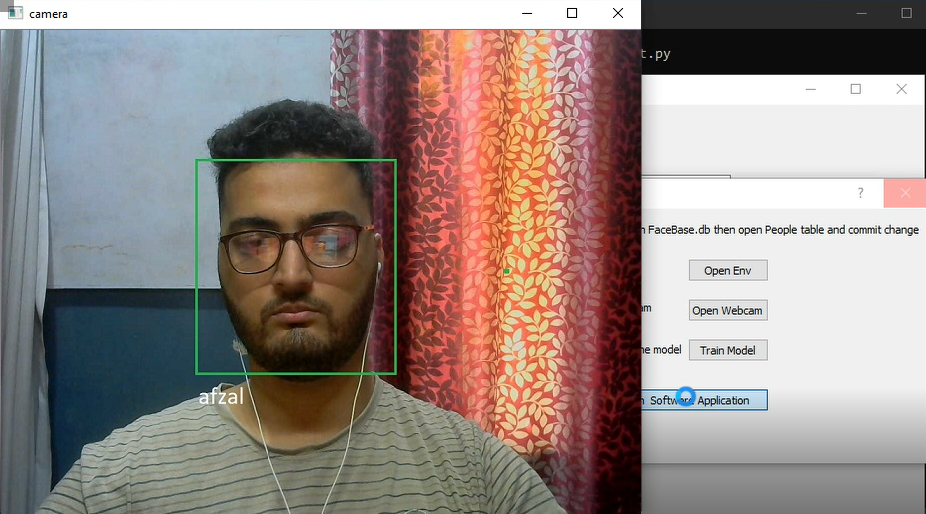
OpenCV provides a training method for pretrained models that can be read using the [cv::CascadeClassifier::load](https://docs.opencv.org/3.4/d1/de5/classcv_1_1CascadeClassifier.html#a1a5884c8cc749422f9eb77c2471958bc) method. The pretrained models are located in the data folder in the OpenCV installation.

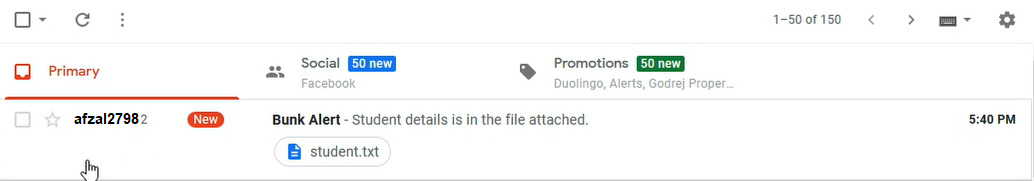
 First, a [cv::CascadeClassifier](https://docs.opencv.org/3.4/d1/de5/classcv_1_1CascadeClassifier.html) is created and the necessary XML file is loaded using the [cv::CascadeClassifier::load](https://docs.opencv.org/3.4/d1/de5/classcv_1_1CascadeClassifier.html#a1a5884c8cc749422f9eb77c2471958bc) method. Afterwards, the detection is done using the [cv::CascadeClassifier::detectMultiScale](https://docs.opencv.org/3.4/d1/de5/classcv_1_1CascadeClassifier.html#aaf8181cb63968136476ec4204ffca498) method, which returns boundary rectangles for the detected faces or eyes.

Then the button Train Model is clicked since it is necessary for the system to be trained with all the photos of students old as well as new to start the system for recognition process.

The trainer/trainer.xml is used for training purposes with faces and their ID’s.

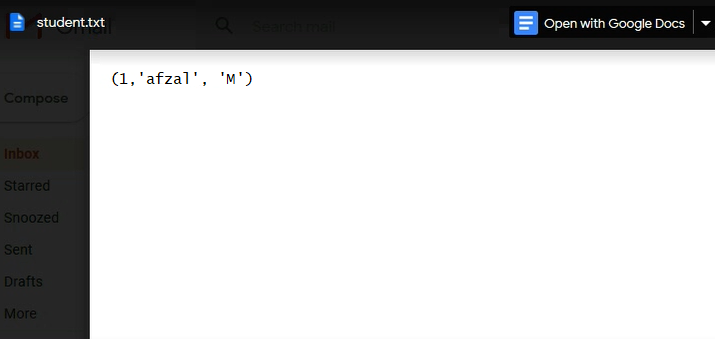
When we click Run Software Application button, the system starts recognizing students in real time through the organization's/college surveillance cameras using the LBPH recognizer module of opencv2. Named as recognizer cv2.face.LBPHFaceRecognizer\_create().

**System working in real-time through a webcam.**



**Email Alert to the Stated Email Address**

When the system captures a student through surveillance camera it sends an email alert to the HOD of that students department. It even sends the students detail in a text file attached with the email which is shown below.



**Attached .txt File in the Email**

Video surveillance using CCTV is now being used everywhere. But effective video surveillance is not implemented anywhere. The current practice shown above uses a webcam to identify student’s faces and send an alert to the HOD of the specified student’s department. It is a proto-type system and well with all the in hand resources and the system is automated it just needs to be started and it does the remaining work of analysing the footages manually which are being captured, and removes some of the hectic work of the security staff. Normally high ended system management can cost many times more on the security staff but at the same cost, we can implement something better and much less all time consuming system for a security guard. Therefore, we have built a alert system which maintains decline in the college corridors

# CHAPTER 5

## 5.1 Conclusion

Face recognition is the most common way of verification and can also be implemented for communication between machines and humans. Face detection and expression recognition is used widely in industries, corporate sectors, financial exchanges for the purpose of security and authentication. In this paper face detection process uses Haar classifiers and produces an accuracy of around 99.2% from still images and 98.7% for video recordings. Face recognition is done using LBPH algorithm and it produces a recognition accuracy of 99.2% for still images and 99.3% from video recordings for the same have been obtained. This work describes a modular system capable of recognizing people in real-time and store their facial photo signature within a minute too. It uses Haar Cascade classifier and CV2 for feature extraction and classification purposes. There is no exact value to detecting the number of faces. The system can be made much more accurate. The performance of face detection can be seen to produce a maximum error of 1.6%, 2.1%, and 0.8% in the case of Real-Time video, image file, and video file respectively which may be considered accurate. The recognition algorithm produces a maximum error of 0.4% which may be considered accurate as well. We have presented an approach for object detection which minimizes computation time while achieving high detection accuracy.

## 5.2 Future Scope

Summing it up in every new face recognition technology renders immense aspects and hopes for future growth. It’s extremely possible that in a couple of years such particles would be able to prepare signs, preparations, motion patterns, palm and ear prints, voice, and scent signature. And most of the cons can be beaten by simple actions from the developer’s side. In future all parameters regarding face recognition will be improved. Detection of faces or images passing by camera will fetch in the form of small slots for better results.

However, because Haar Features have to be determined manually, there is a certain limit to the types of things it can detect. If you give classifier edge and line features, then it will only be able to detect objects with clear edges and lines. Even as a face detector, if we manipulate the face a bit (say, cover up the eyes with sunglasses, or tilt the head to a side), a Haar-based classifier may not be able to recognize the face. A convolutional kernel, on the other hand, has a higher degree of freedom (since it’s determined by training), and could be able to recognize partially covered faces (depending on the quality of the training data). So this part can be improved.

On the plus side, because we don’t need to train Haar-Features, we can create a classifier with a relatively small dataset. All we have to do is train the weightings for each which allows us to train the classifier well without a lot of training images. In addition, it also has a higher execution speed, as Haar-based classifiers typically involve less computations.

# Bibliography

1. Wesley L. Passos, Igor M. Quintanilha, Gabriel M. Araujo “*Real-Time Deep-Learning-Based System for Facial Recognition*” under XXXVI SIMPOSIO BRASILEIRO DE TELECOMUNICAC ̧ OES E PROCESSAMENTO DE SINAIS - SBrT2018, 16-19 DE SETEMBRO DE 2018, CAMPINA GRANDE, PB
2. Paul Viola, Michael Jones “*Rapid Object Detection using a Boosted Cascade of Simple Features*” under ACCEPTED CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION 2001.
3. Florian Schroff, Dmitry Kalenichenko, James Philbin “*FaceNet: A Unified Embedding for Face Recognition and Clustering*”
4. Tadas Baltrusaitis ˇ, Peter Robinson, Louis-Philippe Morency “*OpenFace: an open source facial behavior analysis toolkit*”
5. Muhammad Imran Razzak, Saeeda Naz and Ahmad Zaib “*Deep Learning for Medical Image Processing: Overview, Challenges and Future*”
6. Yaniv Taigman. Ming Yang, Marc’Aurelio Ranzato, Lior Wolf “*DeepFace: Closing the Gap to Human- Level Performance in Face Verification*”
7. Chung-Hua Chu and Yu-Kai Feng “*Study of Eye Blinking to Improve Face Recognition for Screen Unlock on Mobile Devices*” under J Electr Eng Technol.2017; 13(?): 1921-718
8. Bienvenido B. Abad, Jr. “*PROPOSED IMAGE PRE-PROCESSING TECHNIQUES FOR FACE RECOGNITION USING OPENCV*”
9. Dr. Priya Gupta, Nidhi Saxen, Meetika Sharma, Jagriti Tripathia “*Deep Neural Network for Human Face Recognition*” under I.J. Engineering and Manufacturing, 2018, 1, 63-71
10. I. Gallo, S. Nawaz, A. Calefati, G. Piccoli “*TAKING ROLL: A PIPELINE FOR FACE RECOGNITION*”
11. Navneet Dalal and Bill Triggs “*Histograms of Oriented Gradients for Human Detection”*

Shalini SinghMd, Golam Mohiuddin, Swati Pandit, Amogh Banerjee ,Indrasom Gangopadhyay, Indrajit Das *“Face Detection and Recognition by Haar Cascade Classifier , Eigen Face and LBP Histogram*