**Smart Attendance using Deep Learning and Computer Vision**

**A Major Project Report**

**Submitted by**

**S. VIVEK 168W1A0445**

**P. AKHIL KUMAR 168W1A0437**

**B. PAVAN KALYAN 168W1A0406**

In partial fulfilment of the requirements for award of the degree of

**BACHELOR OF TECHNOLOGY**

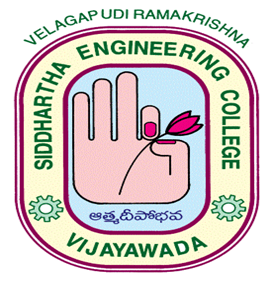
With specialization in

**ELECTRONICS AND COMMUNICATION ENGINEERING**

Under the esteemed guidance of

**Dr. K. SRI RAMA KRISHNA,**

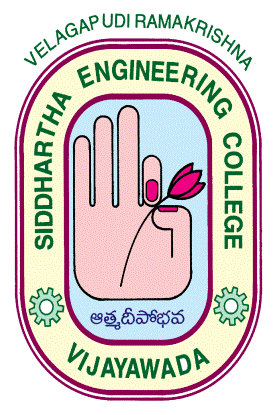
**PROFESSOR, HEAD OF THE DEPARTMENT**

****

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING   
 V. R. SIDDHARTHA ENGINEERING COLLEGE (Autonomous)  
 VIJAYAWADA 520 007**

**March 2020**

**Department of Electronics and Communication Engineering**



**CERTIFICATE**

This is to certify that the Major Project titled **“Smart Attendance Using Deep Learning and Computer Vision”** was prepared and presented by **S.VIVEK (168W1A0445), P AKHIL KUMAR (168W1A0437), B PAVAN KALYAN (168W1A0406)** of B.Tech., 8th Semester, Electronics and Communication Engineering in partial fulfilment of requirements for award of the Degree of Bachelor of Technology in Electronics and Communication Engineering under the Jawaharlal Nehru Technological University Kakinada, Kakinada during the year 2019-20.

**MAJOR PROJECT GUIDE & HEAD OF THE DEPARTMENT**

**(Dr. K. SRI RAMA KRISHNA)**

DATE:

**ACKNOWLEDGEMENT**

We would like to articulate our profound gratitude and indebtedness to our guide **Dr. K. SRI RAMA KRISHNA, Professor, Head of the Department** who has always been a constant motivation and guiding factor throughout the Major Project time in and out as well. It has been a great pleasure for us to get an opportunity to work under his guidance and complete the Major Project successfully.

We sincerely thank our principal **Dr. A. V. Ratna Prasad** garu, for his encouragement during the course of the Major Project.

We express our heartfelt gratitude to our Major Project Co-ordinator who helped us in all aspects.

We thank one and all who have rendered help to us directly or indirectly in the completion of work.

**168W1A0445 S. VIVEK**

**168W1A0437 P. AKHIL KUMAR**

**168W1A0406 B. PAVAN KALYAN**

**DECLARATION**

We hereby declare that the work being presented in this Major Project **“Smart Attendance Using Deep Learning and Computer Vision.”** Submitted towards the partial fulfilment of requirements for the award of the degree of Bachelor of Technology in Electronics and Communication Engineering in V. R. Siddhartha Engineering College, Vijayawada is an authentic record of our work carried out under the supervision of **Dr. K. SRI RAMA KRISHNA, Professor, Head of the ECE Department**, in V. R. Siddhartha Engineering College, Vijayawada. The matter embodied in this dissertation report has not been submitted by us for the award of any other degree. Furthermore, the technical details furnished in various chapters of this report are purely relevant to the above **Major Project**.

**168W1A0445 S. VIVEK**

**168W1A0437 P.AKHIL KUMAR**

**168W1A0406 B. PAVAN KALYAN**

**Table of Contents**

**[1 ABSTRACT 1](#_3znysh7)**

**[2 INTRODUCTION 2](#_2et92p0)**

**[2.1 Object detection 2](#_tyjcwt)**

**[2.2 Dataset and Neural network 2](#_3dy6vkm)**

**[2.3 Deep learning 4](#_1t3h5sf)**

**[2.4 FACE DETECTION 6](#_17dp8vu)**

**[2.4.1 Face Detection Methods: 7](#_3rdcrjn)**

[**2.5**](#_35nkun2) **Face Recognition: 9**

**[2.6 Hardware 13](#_z337ya)**

[**2.7**](#_1y810tw) **Software 16**

**[2.8 Applications 20](#_2xcytpi)**

**[3 LITERATURE SURVEY 21](#_1ci93xb)**

[**4**](#_3whwml4) **ALGORITHMS USED 24**

[**4.1**](#_2bn6wsx) **Haar Cascades 24**

**[4.2 Facenet 25](#_1pxezwc)**

[**5**](#_3o7alnk) **Our Solution 27**

[**6**](#_ihv636) **RESULTS 30**

**[6.1 FACE DETECTION USING HAAR CASCADES 30](#_32hioqz)**

[**6.2**](#_41mghml) **FACE RECOGNITION USING FACENET 30**

**[7 CONCLUSION 31](#_3fwokq0)**

**[8 Bibliography 33](#_4f1mdlm)**

**LIST OF FIGURES**

[Figure 1: Artificial Neural Network Model Diagram 4](#_4d34og8)

[Figure 2 : Algorithms Performance Comparison 5](#_2s8eyo1)

[Figure 3: Different types of Face Detection Methods 7](#_26in1rg)

[Figure 4: Template Matching 8](#_lnxbz9)

[Figure 5: Dimension reduction from 3D to 2D image 10](#_1ksv4uv)

[Figure 6: Deep Neural Network for Face Recognition 12](#_44sinio)

[Figure 7: Neural Networks with Gabor filters 12](#_2jxsxqh)

[Figure 8: Raspberry Pi Figure 16](#_3j2qqm3)

[Figure 9: Anaconda Distribution 19](#_4i7ojhp)

[Figure 10: Haar Like Features 22](#_qsh70q)

[Figure 11: Using Haar Features 22](#_3as4poj)

[Figure 12: Formula for finding the Euclidean distance between points p and q 23](#_49x2ik5)

[Figure 13: Training of a Facenet model 24](#_2p2csry)

[Figure 14: Prototype Deployment Block Diagram 26](#_23ckvvd)

[Figure 15: Detected face using Haar cascades 27](#_1hmsyys)

[Figure 16: Face recognition for multiple faces 28](#_2grqrue)

[Figure 17: Command prompt displaying the names of students present 28](#_vx1227)

**ABBREVIATIONS**

1. CNN: Convolution Neural Networks
2. ANN: Artificial Neural Networks
3. DNN: Deep Neural Network
4. GPU: Graphics Processing Unit
5. VOC: Volatile Organic Compounds
6. VGG: Visual Geometry Group

# ABSTRACT

Attendance is an important part of daily classroom evaluation. Traditional classroom follows a manual attendance marking system i.e. calling a student’s names or by forwarding an attendance sheet. This process is both time consuming and error prone i.e. student proxy etc. Hence we propose a face recognition based smart classroom attendance management system using computer vision and deep learning. We propose to mount a camera at the entrance of the classroom so that the students are visible while entering the class. We then use face detection algorithm followed by face recognition and then mark the attendance of the detected student.

***KEYWORDS****:* Deep learning, Smart classroom, Facenet, Convolutional Neural Networks (CNN), HAAR CASCADES.

# INTRODUCTION

## Object detection

In recent years, with the rapid development of deep learning, a number of research areas have achieved good results, and accompanied by the continuous improvement of convolution neural networks, computer vision has arrived at a new peak. From the ALexNet in 2012 years to the ZF Net in 2013 years, and then to the VGG Net, the ResNet and so on, the architecture of convolutional neural networks is constantly improving. In addition, the return of the convolution neural network also makes the application of computer vision greatly improve, such as face recognition, object detection, object tracking, semantic segmentation, and so on.

Object detection as one of the important applications in the field of computer vision has been the focus of research, and convolution neural network has made great progress in object detection. Object detection is developing from the single object recognition to the multi-object recognition. The meaning of the first is just from an image to identify a single object, it can be said that it is a problem of classification, and the meaning of the later is not only can identify all the objects in an image, including the exact location of the objects. Deep learning has formed a mainstream object recognition algorithm based on RCNN, and these algorithms is refreshing the higher accuracy in a number of famous datasets.

## Dataset and Neural network

For deep learning, dataset and neural network are two important parts. The dataset is the fuel for deep learning so that the number and quality of the dataset will affect the accuracy of the neural network output, and the choice of neural network or the network architecture will also affect the accuracy.

**A. Dataset**

Dataset is one of the foundations of deep learning, for many researchers to get enough data to carry out the experiment just by themselves is a big problem, so we need a lot of open source dataset for everyone to use. Some commonly used datasets in computer vision are the following. The term data set may also be used more loosely, to refer to the data in a collection of closely related tables, corresponding to an experiment or event. Less used names for this kind of data set are data corpus and data stock. An example of this type is the data sets collected by space agencies performing experiments with instruments aboard space probes. Data sets that are so large that traditional data processing applications are inadequate to deal with them are known as big data. In the open data discipline, data set is the unit to measure the information released in a public open data repository. The European Open Data portal aggregates more than half a million data sets. In this field other definitions have been proposed but currently there is not an official one. Some other issues (real-time data sources, non-relational data sets, etc.) increases the difficulty to reach a consensus about it.

**1) ImageNet**

The Imagenet dataset has more than 14 million images covering more than 20,000 categories. There are more than a million pictures with explicit class annotations and annotations of object locations in the image. The Imagenet dataset is one of the most widely used datasets in the field of deep learning. Most of the research work such as image classification, location, and detection are based on this dataset. The Imagenet dataset is detailed and is very easy to use. It is very widely used in the field of computer vision research and has become the "standard" dataset of the current deep learning of image domain to test algorithm performance. There is a well-known challenge called "ImageNet International Computer Vision Challenge" (ILSVRC) based on the Imagenet dataset. It is worth mentioning that the winners of ILSVRC2016 are Chinese teams for all projects. The database was presented for the first time as a poster at the 2009 Conference on Computer Vision and Pattern Recognition (CVPR) in Florida by researchers from the computer Science department at Princeton University. ImageNet primary researchers and inventors include Stanford University computer science professor and researcher Fei-Fei Li.

**2) PASCAL VOC**

The PASCAL VOC (pattern analysis, statistical modelling and computational learning visual object classes) provides standardized image data sets for object class recognition and

provides a common set of tools for accessing the data sets and annotations. The PASCAL VOC dataset includes 20 classes and has a challenge based on this dataset. The PASCAL VOC Challenge is no longer available after 2012, but its dataset is of good quality and well-marked and enables evaluation and comparison of different methods. And because the amount of data of the PASCAL VOC dataset is small, compared to the imagenet dataset, very suitable for researchers to test network programs. Our dataset is also created based on the PASCAL VOC dataset standard.

**3) COCO**

COCO (Common Objects in Context) is a new image recognition, segmentation, and captioning dataset, sponsored by Microsoft. COCO dataset has more than 300,000 images covering 80 object categories. The open source of this dataset makes great progress in semantic sgmentation in recent years, and it has become a "standard" dataset for the performance of image semantic understanding, and also COCO has its own challenge.

**B. Neural Network**

Deep learning used by the network has been constantly improving, in addition to the changes in the network structure, the more is to do some tune based on the original network or apply some trick to make the network performance to enhance. The more well-known algorithms of object detection are a series of algorithms based on R-CNN, mainly in the following

• R-CNN

• SPP-Net

• Fast R-CNN

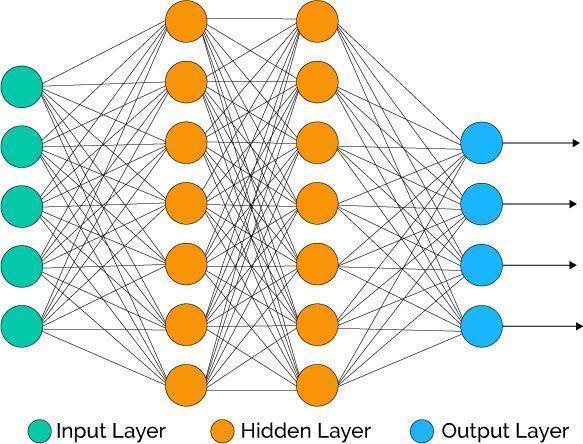
• Faster R-CNN

• YOLO

A more detailed information on the Neural Network used is in given in the section 2.3 below

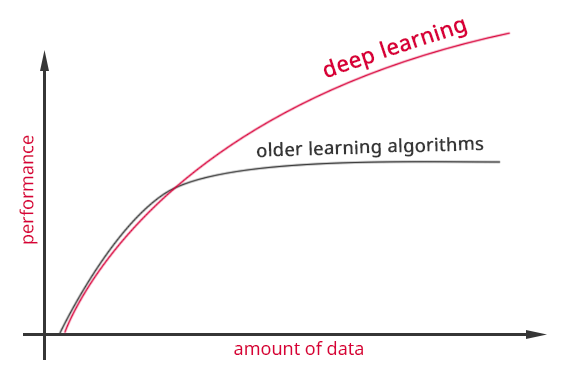
## Deep learning

Deep learning is a sub-field of machine learning dealing with algorithms inspired by the structure and function of the brain called artificial neural networks. In other words, it mirrors the functioning of our brains. Deep learning algorithms are similar to how nervous system structured where each neuron connected each other and passing information.



**Figure 1: Artificial Neural Network Model Diagram**

The history of deep learning dates back to 1943 when Warren McCulloch and Walter Pitts created a computer model based on the neural networks of the human brain. Warren McCulloch and Walter Pitts used a combination of mathematics and algorithms they called threshold logic to mimic the thought process. Since then, deep learning has evolved steadily, over the years with two significant breaks in its development. The development of the basics of a continuous Back Propagation Model is credited to Henry J. Kelley in 1960. Stuart Dreyfus came up with a simpler version based only on the chain rule in 1962. The concept of back propagation existed in the early 1960s but only became useful until 1985. Deep learning models tend to perform well with amount of data where as old machine learning models stops improving after a saturation point.



**Figure 2 : Algorithms Performance Comparison**

A deep neural network (DNN) is an artificial neural network (ANN) with multiple layers between the input and output layers. The DNN finds the correct mathematical manipulation to turn the input into the output, whether it be a linear relationship or a non-linear relationship. The network moves through the layers calculating the probability of each output. For example, a DNN that is trained to recognize dog breeds will go over the given image and calculate the probability that the dog in the image is a certain breed. The user can review the results and select which probabilities the network should display (above a certain threshold, etc.) and return the proposed label. Each mathematical manipulation as such is considered a layer, and complex DNN have many layers, hence the name "deep" networks. The goal is that eventually, the network will be trained to decompose an image into features, identify trends that exist across all samples and classify new images by their similarities without requiring human input. DNNs can model complex non-linear relationships. DNN architectures generate compositional models where the object is expressed as a layered composition of primitives. The extra layers enable composition of features from lower layers, potentially modelling complex data with fewer units than a similarly performing shallow network. Deep architectures include many variants of a few basic approaches. Each architecture has found success in specific domains. It is not always possible to compare the performance of multiple architectures, unless they have been evaluated on the same data sets. DNNs are typically feedforward networks in which data flows from the input layer to the output layer without looping back. At first, the DNN creates a map of virtual neurons and assigns random numerical values, or "weights", to connections between them. The weights and inputs are multiplied and return an output between 0 and 1. If the network didn’t accurately recognize a particular pattern, an algorithm would adjust the weights. That way the algorithm can make certain parameters more influential, until it determines the correct mathematical manipulation to fully process the data. Recurrent neural networks (RNNs), in which data can flow in any direction, are used for applications such as language modelling. Long short-term memory is particularly effective for this use. Convolutional deep neural networks (CNNs) are used in computer vision. CNNs also have been applied to acoustic modelling for automatic speech recognition (ASR). As with ANNs, many issues can arise with naively trained DNNs. Two common issues are overfitting and computation time.

DNNs are prone to overfitting because of the added layers of abstraction, which allow them to model rare dependencies in the training data. Regularization methods such as Ivakhnenko's unit pruning or weight decay or sparsity can be applied during training to combat overfitting. Alternatively dropout regularization randomly omits units from the hidden layers during training. This helps to exclude rare dependencies. Finally, data can be augmented via methods such as cropping and rotating such that smaller training sets can be increased in size to reduce the chances of overfitting. DNNs must consider many training parameters, such as the size (number of layers and number of units per layer), the learning rate, and initial weights. Sweeping through the parameter space for optimal parameters may not be feasible due to the cost in time and computational resources. Various tricks, such as batching (computing the gradient on several training examples at once rather than individual examples) speed up computation. Large processing capabilities of many-core architectures (such as, GPUs or the Intel Xeon Phi) have produced significant speedups in training, because of the suitability of such processing architectures for the matrix and vector computations. Alternatively, engineers may look for other types of neural networks with more straightforward and convergent training algorithms. CMAC (cerebellar model articulation controller) is one such kind of neural network. It doesn't require learning rates or randomized initial weights for CMAC. The training process can be guaranteed to converge in one step with a new batch of data, and the computational complexity of the training algorithm is linear with respect to the number of neurons involved.

## FACE DETECTION

In the past few years, face recognition has received significant consideration and is considered as one of the most promising applications in the field of image analysis. Face detection can consider a substantial part of face recognition operations. According to its strength to focus computational resources on the section of an image holding a face. The method of face detection in pictures is complicated because of variability present across human faces such as pose, expression, position and orientation, skin colour, the presence of glasses or facial hair, differences in camera gain, lighting conditions, and image resolution.

Object detection is one of the computer technologies, which connected to the image processing and computer vision and it interacts with detecting instances of an object such as human faces, building, tree, car, etc. The primary aim of face detection algorithms is to determine whether there is any face in an image or not.

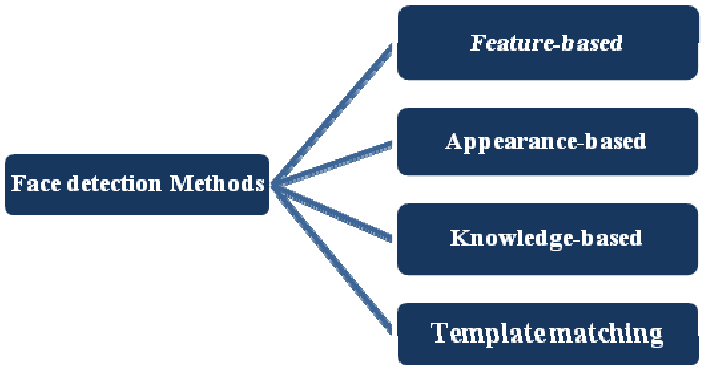
In recent times, a lot of study work proposed in the field of Face Recognition and Face Detection to make it more advanced and accurate, but it makes a revolution in this field when Viola-Jones comes with its Real-Time Face Detector, which is capable of detecting the faces in real-time with high accuracy.

Face Detection is the first and essential step for face recognition, and it is used to detect faces in the images. It is a part of object detection and can use in many areas such as security, bio-metrics, law enforcement, entertainment, personal safety, etc.

It is used to detect faces in real time for surveillance and tracking of person or objects. It is widely used in cameras to identify multiple appearances in the frame Ex- Mobile cameras and DSLR’s. Facebook is also using face detection algorithm to detect faces in the images and recognise them.

### **Face Detection Methods:**

Yan, Kriegman, and Ahuja presented a classification for face detection methods. These methods divided into four categories, and the face detection algorithms could belong to two or more groups. These categories are as follows-



**Figure 3: Different types of Face Detection Methods**

**Different types of Face Detection Methods**

**1.Knowledge-Based:**

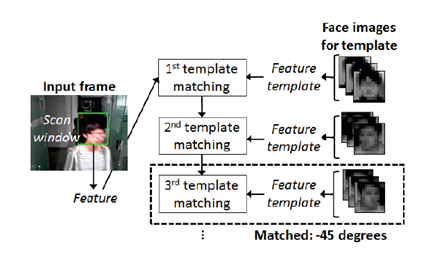
The knowledge-based method depends on the set of rules, and it is based on human knowledge to detect the faces. Ex- A face must have a nose, eyes, and mouth within certain distances and positions with each other. The big problem with these methods is the difficulty in building an appropriate set of rules. There could be many false positive if the rules were too general or too detailed. This approach alone is insufficient and unable to find many faces in multiple images.

**2.Feature-Based:**

The feature-based method is to locate faces by extracting structural features of the face. It is first trained as a classifier and then used to differentiate between facial and non-facial regions. The idea is to overcome the limits of our instinctive knowledge of faces. This approach divided into several steps and even photos with many faces they report a success rate of 94%.

**3.Template Matching:**

Template Matching method uses pre-defined or parameterised face templates to locate or detect the faces by the correlation between the templates and input images. Ex- a human face can be divided into eyes, face contour, nose, and mouth. Also, a face model can be built by edges just by using edge detection method. This approach is simple to implement, but it is inadequate for face detection. However, deformable templates have been proposed to deal with these problems.



**Figure 4: Template Matching**

**4. Appearance-Based:**

The appearance-based method depends on a set of delegate training face images to find out face models. The appearance-based approach is better than other ways of performance. In general, appearance-based methods rely on techniques from statistical analysis and machine learning to find the relevant characteristics of face images. This method is also used in feature extraction for face recognition.

The appearance-based model is further divided into sub-methods for the use of face detection which are as follows-

**4.1 Eigenface-Based:**

Eigenface based algorithm used for Face Recognition, and it is a method for efficiently representing faces using Principal Component Analysis.

**4.2 Distribution-Based:**

The algorithms like PCA and Fisher’s Discriminant can be used to define the subspace representing facial patterns. There is a trained classifier, which correctly identifies instances of the target pattern class from the background image patterns.

**4.3 Neural-Networks:**

Many detection problems like object detection, face detection, emotion detection, and face recognition, etc. have been faced successfully by Neural Networks.

**4.4 Support Vector Machine:**

Support Vector Machines are linear classifiers that maximise the margin between the decision hyperplane and the examples in the training set. Osuna et al. first applied this classifier to face detection.

**4.5 Naive Bayes Classifiers:**

They computed the probability of a face to be present in the picture by counting the frequency of occurrence of a series of the pattern over the training images. The classifier captured the joint statistics of local appearance and position of the faces.

## Face Recognition:

Face Recognition is a recognition technique used to detect faces of individuals whose images saved in the data set. Despite the point that other methods of identification can be more accurate, face recognition has always remained a significant focus of research because of its non-meddling nature and because it is people’s facile method of personal identification.

**Face Recognition Methods:**

There are different methods for face recognition, which are as follows-

**1. Geometric Based / Template Based:**

Face recognition algorithms classified as geometry based or template based algorithms. The template-based methods can be constructed using statistical tools like SVM [Support Vector Machines], PCA [Principal Component Analysis], LDA [Linear Discriminant Analysis], Kernel methods or Trace Transforms. The geometric feature-based methods analyse local facial features and their geometric relationship. It is also known as a feature-based method.

**2. Piecemeal / Wholistic:**

The relation between the elements or the connection of a function with the whole face not undergone into the amount, many researchers followed this approach, trying to deduce the most relevant characteristics. Some methods attempted to use the eyes, a combination of features and so on. Some Hidden Markov Model methods also fall into this category, and feature processing is very famous in face recognition.

**3. Appearance-Based / Model-Based:**

The appearance-based method shows a face regarding several images. An image considered as a high dimensional vector. This technique is usually used to derive a feature space from the image division. The sample image compared to the training set. On the other hand, the model-based approach tries to model a face. The new sample implemented to the model and the parameters of the model used to recognise the image.

The appearance-based method can classify as linear or nonlinear. Ex- PCA, LDA, IDA used in direct approach whereas Kernel PCA used in nonlinear approach. On the other hand, in the model-based method can be classified as 2D or 3D Ex- Elastic Bunch Graph Matching used.

**4. Template / Statistical / Neural Networks Based:**

**4.1. Template Matching:**

In template matching the patterns are represented by samples, models, pixels, textures, etc. The recognition function is usually a correlation or distance measure.

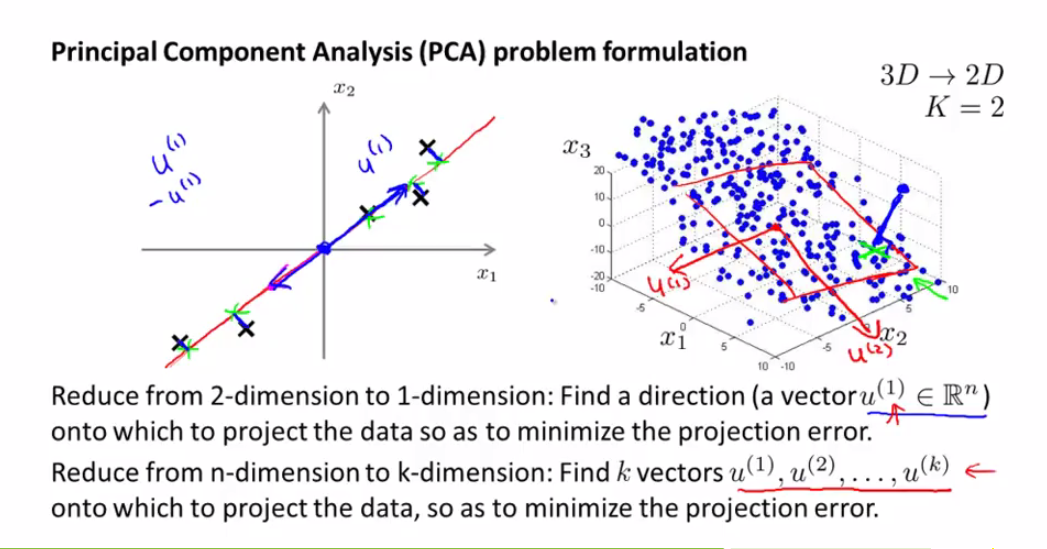
**4.2. Statistical Approach:**

In the Statistical approach, the patterns expressed as features. The recognition function in a discriminant function. Each image represents features. Therefore, the goal is to choose and apply the right statistical tool for extraction and analysis.

There are many statistical tools, which are used for face recognition. These analytical tools are used in two or more groups or classification methods. These tools are as follows-

**4.2.1. Principal Component Analysis [PCA]:**

One of the most used and cited statistical method is the Principal Component Analysis. A mathematical procedure performs a dimensionality reduction by extracting the principal component of multi-dimensional data.



**Figure 5: Dimension reduction from 3D to 2D image**

**4.2.2. Discrete Cosine Transform [DCT]:**

It signifies a series of data points regarding a sum of cosine functions with different oscillating frequencies. The Discrete Cosine Transform is based on Fourier discrete transform and therefore, by compacting the variations it can be used to transform images and allowing an efficient dimensionality reduction.

**4.2.3. Linear Discriminant Analysis [LDA]:**

LDA is widely used to find the linear combination of features while preserving class separability. Unlike PCA, the LDA tries to model to the difference between levels. For each level the LDA obtains differences in multiple projection vectors.

**4.2.4. Locality Preserving Projections [LPP]:**

HE and NIYOGI introduced The LPP. It is the best alternative of PCA for preserve locality structure and designing. Pattern recognition algorithms usually search for the nearest pattern or neighbours. Therefore, the locality maintaining the quality of LLP can quicken the recognition.

**4.2.5. Gabor Wavelet:**

In this algorithm, it signifies that Neuro-physiological data evidence from the visual cortex of mammalian brains suggests that simple cells in the visual cortex can view as a family of self-similar 2D Gabor wavelets. The Gabor functions proposed by Daugman are local spatial bandpass filters that achieve the theoretical limit for conjoint resolution of information in the 2D spatial and 2D Fourier domains.

**4.2.6. Independent Component Analysis [ICA]:**

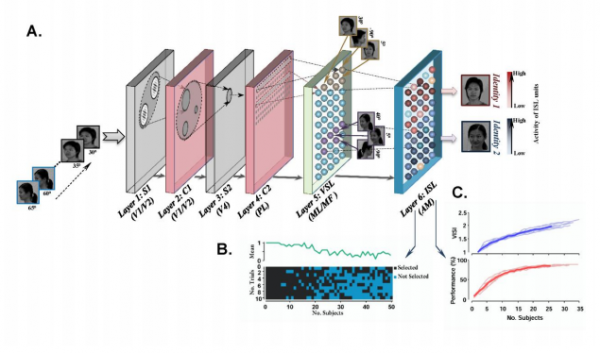
ICA aims to transform the data as linear combinations of the statistically independent data point. Therefore, its goal is to provide an independent instead that uncorrelated image representation. ICA is an alternative to PCA, which give a more powerful data representation. It is a discriminant analysis criterion, which can be used to enhance PCA.

**4.2.7. Kernel PCA:**

Scholkopf et al. introduced the use of Kernel functions for performing nonlinear PCA. Its basic methodology is to apply a nonlinear mapping to the input and then solve a linear PCA in the resulting feature subspace.

**4.3. Neural Networks:**

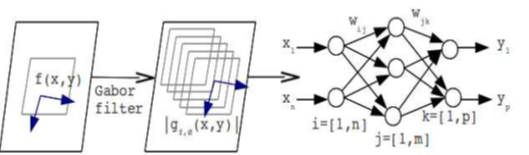
Neural Network has continued to use pattern recognition and classification. Kohonen was the first to show that a neuron network could be used to recognise aligned and normalised faces. There are methods, which perform feature extraction using neural networks. There are many methods, which combined with tools like PCA or LCA and make a hybrid classifier for face recognition. These are like Feed Forward Neural Network with additional bias, Self-Organizing Maps with PCA, and Convolutional Neural Networks with multi-layer perception, etc. These can increase the efficiency of the models.



**Figure 6: Deep Neural Network for Face Recognition**

**4.3.1. Neural Networks with Gabor Filters:**

The algorithm achieves face recognition by implementing a multilayer perceptron with a back-propagation algorithm. Firstly, there is a pre-processing step. Each image normalised in phases of contrast and illumination. Then each image is processed through a Gabor filter. The Gabor filter has five orientation parameters and three spatial frequencies, so there are 15 Gabor wavelengths.



**Figure 7: Neural Networks with Gabor filters**

**4.3.2. Neural Networks and Hidden Markov Models:**

Hidden Markov Models are a statistical tool used in face recognition. They have used in conjunction with neural networks. It is generated in a neural network that trains pseudo 2D HMM. The input of this 2D HMM process is the output of the ANN, and It provides the algorithm with the proper dimensionality reduction.

**4.3.3. Fuzzy Neural Networks:**

The fuzzy neural networks for face recognition introduce in 2009. In this a face recognition system using a multilayer perceptron. The concept behind this approach is to capture decision surfaces in nonlinear manifolds a task that a simple MLP can hardly complete. The feature vectors are obtained using Gabor wavelength transforms.

**Dlib library**

Dlib is a general purpose [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) software [library](https://en.wikipedia.org/wiki/Library_(computing)) written in the programming language [C++](https://en.wikipedia.org/wiki/C%2B%2B). Its design is heavily influenced by ideas from [design by contract](https://en.wikipedia.org/wiki/Design_by_contract) and [component-based software engineering](https://en.wikipedia.org/wiki/Component-based_software_engineering). Thus, it is the first and foremost set of independent software components. It is [open-source software](https://en.wikipedia.org/wiki/Open-source_software) released under a [Boost Software License](https://en.wikipedia.org/wiki/Boost_(C%2B%2B_libraries)#License).

Since development began in 2002, Dlib has grown to include a wide variety of tools. As of 2016, it contains software components for dealing with [networking](https://en.wikipedia.org/wiki/Computer_network), [threads](https://en.wikipedia.org/wiki/Thread_(computing)), [graphical user interfaces](https://en.wikipedia.org/wiki/Graphical_user_interface), [data structures](https://en.wikipedia.org/wiki/Data_structure), [linear algebra](https://en.wikipedia.org/wiki/Linear_algebra), [machine learning](https://en.wikipedia.org/wiki/Machine_learning), [image processing](https://en.wikipedia.org/wiki/Image_processing), [data mining](https://en.wikipedia.org/wiki/Data_mining), [XML](https://en.wikipedia.org/wiki/XML) and text parsing, [numerical optimization](https://en.wikipedia.org/wiki/Numerical_optimization), [Bayesian networks](https://en.wikipedia.org/wiki/Bayesian_network), and many other tasks. In recent years, much of the development has been focused on creating a broad set of statistical machine learning tools and in 2009 Dlib was published in the [Journal of Machine Learning Research](https://en.wikipedia.org/wiki/Journal_of_Machine_Learning_Research). Since then it has been used in a wide range of domains.

## Hardware

Hardware plays a major role and is completely responsible for the best and sometimes the worse results from our program designed for an application. Machine Learning being one of the advanced ways of computing requires a great computation power that lead a path towards the usage of GPU’s.

**Why GPU?**

A graphics processing unit (GPU) is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device. GPUs are used in embedded systems, mobile phones, personal computers, workstations, and game consoles. Modern GPUs are very efficient at manipulating computer graphics and image processing. Their highly parallel structure makes them more efficient than general-purpose CPUs for algorithms that process large blocks of data in parallel. In a personal computer, a GPU can be present on a video card or embedded on the motherboard. In certain CPUs, they are embedded on the CPU die. So higher the capacity of the GPU higher the computation and Greater will be the performance of the application.

Many companies have produced GPUs under several brand names. In 2009, Intel, Nvidia and AMD/ATI were the market share leaders, with 49.4%, 27.8% and 20.6% market share respectively. However, those numbers include Intel's integrated graphics solutions as GPUs. Not counting those, Nvidia and AMD control nearly 100% of the market as of 2018. Their respective market shares are 66% and 33%. In addition, S3 Graphics and Matrox produce GPUs. Modern smartphones are also using mostly Adreno GPUs from Qualcomm, PowerVR GPUs from Imagination Technologies and Mali GPUs from ARM.

Modern GPUs use most of their transistors to do calculations related to 3D computer graphics. In addition to the 3D hardware, today's GPUs include basic 2D acceleration and framebuffer capabilities (usually with a VGA compatibility mode). Newer cards like AMD/ATI HD5000-HD7000 even lack 2D acceleration; it has to be emulated by 3D hardware. GPUs were initially used to accelerate the memory-intensive work of texture mapping and rendering polygons, later adding units to accelerate geometric calculations such as the rotation and translation of vertices into different coordinate systems. Recent developments in GPUs include support for programmable shaders which can manipulate vertices and textures with many of the same operations supported by CPUs, oversampling and interpolation techniques to reduce aliasing, and very high-precision colour spaces. Because most of these computations involve matrix and vector operations, engineers and scientists have increasingly studied the use of GPUs for non-graphical calculations; they are especially suited to other embarrassingly parallel problems. With the emergence of deep learning, the importance of GPUs has increased. In research done by Indigo, it was found that while training deep learning neural networks, GPUs can be 250 times faster than CPUs. The explosive growth of Deep Learning in recent years has been attributed to the emergence of general-purpose GPUs. There has been some level of competition in this area with ASICs, most prominently the Tensor Processing Unit (TPU) made by Google. However, these can require changes to existing code and GPUs are still very popular.

Most GPUs made since 1995 support the YUV colour space and hardware overlays, important for digital video playback, and many GPUs made since 2000 also support MPEG primitives such as motion compensation and iDCT. This process of hardware accelerated video decoding, where portions of the video decoding process and video post-processing are offloaded to the GPU hardware, is commonly referred to as "GPU accelerated video decoding", "GPU assisted video decoding", "GPU hardware accelerated video decoding" or "GPU hardware assisted video decoding"

For Simple applications with confined space we can use a simple system with decent computation power. So, for such application we can go for a Raspberry Pi Computer which is small and not less than a normal PC in performance. Several generations of Raspberry Pi’s have been released. All models feature a Broadcom system on a chip (SoC) with an integrated ARM-compatible central processing unit (CPU) and on-chip graphics processing unit (GPU).

**Raspberry Pi:**

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote teaching of basic computer science in schools and in developing countries. The original model became far more popular than anticipated,[8] selling outside its target market for uses such as robotics. It does not include peripherals (such as keyboards and mice) and cases. However, some accessories have been included in several official and unofficial bundles.

The organisation behind the Raspberry Pi consists of two arms. The first two models were developed by the Raspberry Pi Foundation. After the Pi Model B was released, the Foundation set up Raspberry Pi Trading, with Eben Upton as CEO, to develop the third model, the B+. Raspberry Pi Trading is responsible for developing the technology while the Foundation is an educational charity to promote the teaching of basic computer science in schools and in developing countries.

**Specifications of a Raspberry Pi:**

Processor speed ranges from 700 MHz to 1.4 GHz for the Pi 3 Model B+, on-board memory ranges from 256 MB to 1 GB RAM. Secure Digital (SD) cards in MicroSD form factor are used to store the operating system and program memory. The boards have one to four USB ports. For video output, HDMI and composite video are supported, with a standard 3.5 mm tip-ring-sleeve jack for audio output. Lower-level output is provided by a number of GPIO pins, which support common protocols like I²C. The B-models have an 8P8C Ethernet port and the Pi 3 and Pi Zero W have on-board Wi-Fi 802.11n and Bluetooth.

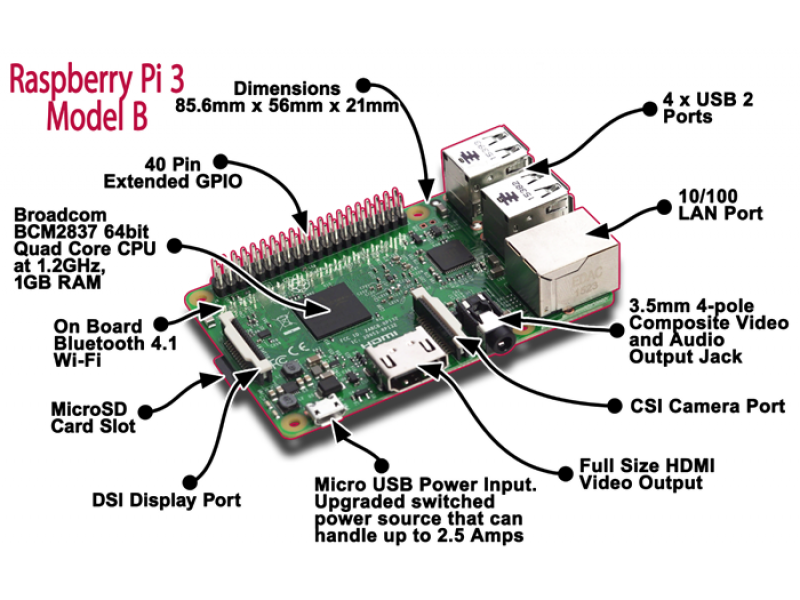
Most Raspberry Pi systems-on-chip could be overclocked to 800 MHz, and some to 1000 MHz. There are reports the Raspberry Pi 2 can be similarly overclocked, in extreme cases, even to 1500 MHz (discarding all safety features and over-voltage limitations). In the Raspbian Linux distro the overclocking options on boot can be done by a software command running "sudo raspi-config" without voiding the warranty.[30] In those cases the Pi automatically shuts the overclocking down if the chip temperature reaches 85 °C (185 °F), but it is possible to override automatic over-voltage and overclocking settings (voiding the warranty); an appropriately sized heat sink is needed to protect the chip from serious overheating.

Newer versions of the firmware contain the option to choose between five overclock ("turbo") presets that when used, attempt to maximise the performance of the SoC without impairing the lifetime of the board. This is done by monitoring the core temperature of the chip and the CPU load, and dynamically adjusting clock speeds and the core voltage. When the demand is low on the CPU or it is running too hot the performance is throttled, but if the CPU has much to do and the chip's temperature is acceptable, performance is temporarily increased with clock speeds of up to 1 GHz, depending on the board version and on which of the turbo settings is used.

GPU & Processing Power:

The GPU provides Open GL ES 2.0, hardware-accelerated Open VG, and 1080p high-profile decode and is capable of 1Gpixel/s, 1.5Gtexel/s or 24 GFLOPs of general-purpose compute. It supports Blu-ray quality video, using H.264 at 40MBits/s so it is a best fit for many applications. In the highest (turbo) present the SDRAM clock was originally 500 MHz, but this was later changed to 600 MHz because 500 MHz sometimes causes SD card corruption. Simultaneously in high mode the core clock speed was lowered from 450 to 250 MHz, and in medium mode from 333 to 250 MHz.

For the new Pi 3, the Broadcom BCM2837 system-on-chip (SoC) includes four high-performance ARM Cortex-A53 processing cores running at 1.2GHz with 32kB Level 1 and 512kB Level 2 cache memory, a Video Core IV graphics processor, and is linked to a 1GB LPDDR2 memory module on the rear of the board.



**Figure 8: Raspberry Pi Figure**

The Raspberry Pi Foundation provides Raspbian, a Debian-based Linux distribution for download, as well as third-party Ubuntu, Windows 10 IoT Core, RISC OS, and specialised media centre distributions. It promotes Python and Scratch as the main programming languages, with support for many other languages. The default firmware is closed source, while an unofficial open source is available. Many other operating systems can also run on the Raspberry Pi, including the formally verified microkernel, seL4. Other third-party operating systems available via the official website include Ubuntu MATE, Windows 10 IoT Core, RISC OS and specialised distributions for the Kodi media centre and classroom management.

## Software

**Programming Language: Python**

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Van Rossum led the language community until stepping down as leader in July 2018. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural. It also has a comprehensive standard library.

Python has top the charts in the recent years over other programming languages like C, C++ and Java and is widely used by the programmers. The language has undergone a drastic change since its release 25 years ago as many add-on features are introduced. The Python 1.0 had the module system of Modula-3 and interacted with Amoeba Operating System with varied functioning tools. Python 2.0 introduced in the year 2000 had features of garbage collector and Unicode Support. Python 3.0 introduced in the year 2008 had a constructive design that avoids duplicate modules and constructs. With the added features, now the companies are using Python 3.5.

The software development companies prefer Python language because of its versatile features and fewer programming codes. Nearly 14% of the programmers use it on the operating systems like UNIX, Linux, Windows and Mac OS. The programmers of big companies use Python as it has created a mark for itself in the software development with characteristic features like-Interactive, Interpreted, Modular, Dynamic, Object-oriented, Portable, High level, Extensible in C++ & C.

The Python language has diversified application in the software development companies such as in gaming, web frameworks and applications, language development, prototyping, graphic design applications, etc. This provides the language a higher plethora over other programming languages used in the industry. It provides large standard libraries that include the areas like string operations, Internet, web service tools, operating system interfaces and protocols. Most of the highly used programming tasks are already scripted into it that limits the length of the codes to be written in Python.

**Integration Feature**

Python integrates the Enterprise Application Integration that makes it easy to develop Web services by invoking COM or COBRA components. It has powerful control capabilities as it calls directly through C, C++ or Java via Jython. Python also processes XML and other mark-up languages as it can run on all modern operating systems through same byte code. The language has extensive support libraries and clean object-oriented designs that increase two to tenfold of programmer’s productivity while using the languages like Java, VB, Perl, C, C++ and C#. With its strong process integration features, unit testing framework and enhanced control capabilities contribute towards the increased speed for most applications and productivity of applications. It is a great option for building scalable multi-protocol network applications.

Python being the simplest and powerful language with many libraries and packages could be used for various applications like Machine Learning, Image Processing, Data Analytics that require a lot of computation and makes it a simple task with a better use of packages and libraries available in the Internet.

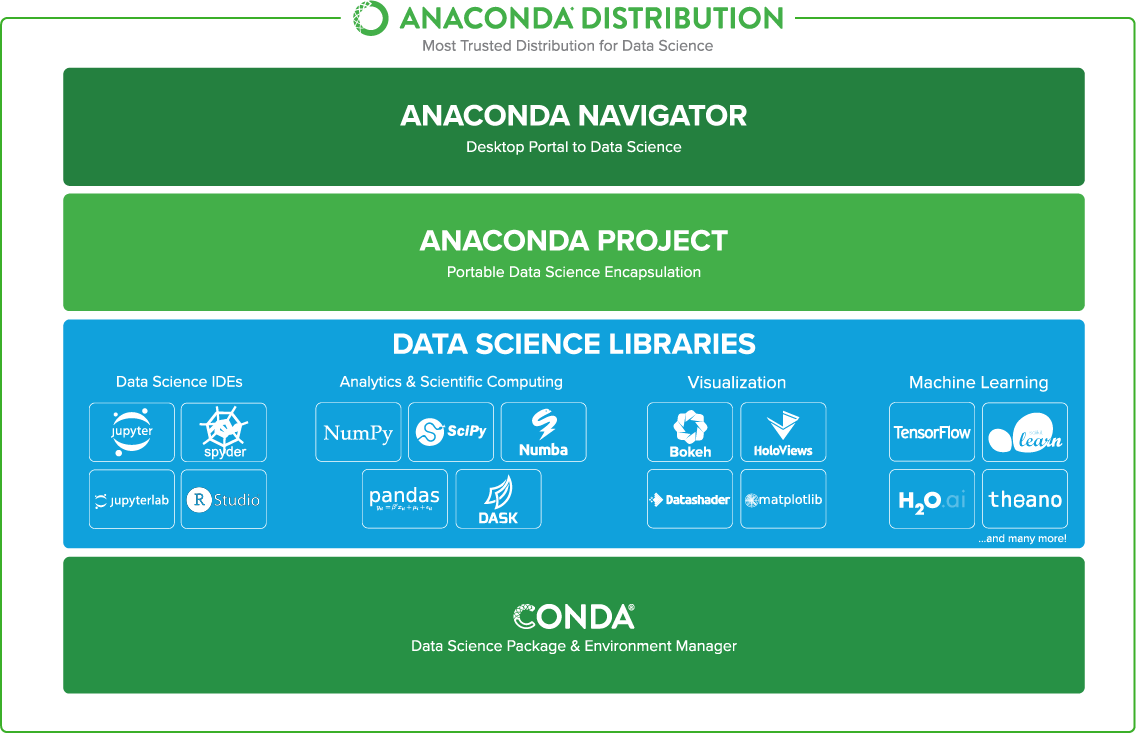
**Integrated Development Environment (IDE): Anaconda**

Object Detection using Deep Neural Networks being an application which involves the use of various libraries and packages developed in Python we can go for a framework called **Anaconda**.

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system Conda. The Anaconda distribution is used by over 12 million users and includes more than 1400 popular data-science packages suitable for Windows, Linux, and MacOS.

Packages generally used for Object Detection in Anaconda Framework:

1. NumPy: NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.
2. Pandas: Pandas is a software library written for the Python programming language for data manipulation and analysis. It offers data structures and operations for manipulating numerical tables and time series. It offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.
3. Scikit-Learn: Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.
4. OpenCV: OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez. OpenCV is released under a BSD license and hence it’s free for both academic and commercial use. It has C++, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform. Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics.
5. TensorFlow: TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library and is also used for machine learning applications such as neural networks. It is used for both research and production at Google. ‍ It is a standard expectation in the industry to have experience in TensorFlow to work in machine learning. It uses a system of multi-layered nodes that allows you to quickly set up, train, and deploy artificial neural networks with large datasets. This is what allows Google to identify objects in photos or understand spoken words in its voice-recognition app. The interesting thing about TensorFlow is that when you write a program in Python, you can compile and run on either your CPU or GPU. So, you don’t have to write at the C++ or CUDA level to run on GPUs.
6. Keras: Keras is an open-source neural-network library written in Python. It can run on top of TensorFlow, Microsoft Cognitive Toolkit, Theano. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System)



**Figure 9: Anaconda Distribution**

## Applications

Object detection could be used in many ways. The Object Detection as a combination with Internet of Things made it an excellent way to solve many real-life problems. Some of them are:

* Weapon and Unusual Activity Detection
* Theft Detection and Burglar Alarm System
* Self-Driving Cars
* Smart Farming and Industrial Automation
* Drone Surveillance and Traffic Monitoring

# LITERATURE SURVEY

[**Kaipeng Zhang**](https://arxiv.org/search/cs?searchtype=author&query=Zhang%2C+K) **et al**. have developed a Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. Face detection and alignment in an unconstrained environment are challenging due to various poses, illuminations and occlusions. Recent studies show that deep learning approaches can achieve impressive performance on these two tasks. In this paper, a deep cascaded multi-task framework which exploits the inherent correlation between them to boost up their performance has been proposed. In particular, this framework adopts a cascaded structure with three stages of carefully designed deep convolutional networks that predict face and landmark location in a coarse-to-fine manner. In addition, in the learning process, a new online hard sample mining strategy that can improve the performance automatically without manual sample selection has been proposed. This method achieves superior accuracy over the state-of-the-art techniques on the challenging FDDB and WIDER FACE benchmark for face detection, and AFLW benchmark for face alignment, while keeping real time performance (Zhang) **.**

[**Florian Schroff**](https://arxiv.org/search/cs?searchtype=author&query=Schroff%2C+F) **et al.** have developed a A Unified Embedding for Face Recognition and Clustering. This paper presents a system, called FaceNet, that directly learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity. Once this space has been produced, tasks such as face recognition, verification and clustering can be easily implemented using standard techniques with FaceNet embeddings as feature vectors. This method uses a deep convolutional network trained to directly optimize the embedding itself, rather than an intermediate bottleneck layer as in previous deep learning approaches. To train, triplets of roughly aligned matching / non-matching face patches generated using a novel online triplet mining method have been used. The benefit of this approach is much greater representational efficiency: state-of-the-art face recognition performance has been achieved using only 128-bytes per face (Schroff).

[**Sujit Kumar Gupta**](https://ieeexplore.ieee.org/author/37086436026) **et al.** proposed a system that uses Max-Margin Face Detection (MMFD) technique for the face detection and the model is trained using the Inception-V3 CNN technique for the students' identification. The proposed smart classroom system was tested for a classroom with 20 students at National Institute of Technology Karnataka Surathkal, Mangalore, India and experimental results demonstrate a train and test accuracy of 97.67% and 96.66% (S. K. Gupta).

[**Refik Samet**](https://ieeexplore.ieee.org/author/37393531300) **et al.** proposed a face recognition-based mobile automatic classroom attendance management system needing no extra equipment. To this end, a filtering system based on Euclidean distances calculated by three face recognition techniques, namely Eigenfaces, Fisherfaces and Local Binary Pattern, has been developed for face recognition. The proposed system includes three different mobile applications for teachers, students, and parents to be installed on their smartphones to manage and perform the real-time attendance-taking process. The proposed system was tested among students at Ankara University, and the results obtained were very satisfactory. (Tanriverdi).

[**Nazare Kanchan Jayant**](https://ieeexplore.ieee.org/author/37085901364) **et al.** proposed an automated Attendance Management System (AMS) based on face detection and face recognition techniques. The system employs modified Viola-Jones algorithm for face detection, and alignment- free partial face recognition algorithm for face recognition. After successful recognition of a student, the system automatically updates the attendance in the excel sheet. The proposed system improves the performance of existing attendance management systems by eliminating manual calling, marking and entry of attendance in institutional websites (Borra).

[**Visar Shehu**](https://ieeexplore.ieee.org/author/37546097000) **et al.** introduced a new approach in automatic attendance management systems, extended with computer vision algorithms. A real time face detection algorithm integrated on an existing Learning Management System (LMS) has been proposed, which automatically detects and registers students attending a lecture. The system represents a supplemental tool for instructors, combining algorithms used in machine learning with adaptive methods used to track facial changes during a longer period of time. This system aims to be less time consuming than traditional methods, at the same time being nonintrusive and not interfere with the regular teaching process. The tool promises to offer accurate results and a more detailed reporting system which shows student activity and attendance in a classroom (Dika).

[**Khem Puthea**](https://ieeexplore.ieee.org/author/37086333217) **et al** reviewed the previous works on the attendance management system based on facial recognition. The literature review on the earlier work or related work has been provided, but also the deep analysis of Principal Component Analysis has been provided, discussion, suggestions for future work (K. Puthea).

[**Aayush Mittal et al**](https://ieeexplore.ieee.org/author/37086379837)**.**presented Cloud Based Intelligent Attendance System through Video Streaming. The architecture and algorithm used in each stage of this system has been described. The system is based on the concept of face recognition relatively on a larger scale. The system identifies and authenticates each student present in the class. Through the proposed system, the attendance of a group of students is carried out at a single point of time. The objective of the system is to automate the traditional way of taking the attendance on registers and to integrate the system with the cloud so as to make all the records readily available that are maintained by the system by reducing the errors. Thus, when compared with the other alternatives of marking the attendance, this system proves to be more reliable and accurate (A. Mittal).

[**Jose Edwin**](https://ieeexplore.ieee.org/author/37086854350)**et al** proposed a Real time monitoring of public places for possible suspects. The manual labour involved and the human errors that can occur makes the system less efficient. The authors presented the implementation of an intelligent multi camera Face Recognition based surveillance system using FaceNet and MTCNN algorithm on Jetson TX2. The proposed portable system tracks the subject or the suspect with the camera ID/location together with the timestamp and logs his presence in the database, using multiple camera installations (J. Edwin).

**Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alex Alemi et al.** observed that very deep convolutional networks have been central to the largest advances in image recognition performance in recent years. One example is the Inception architecture that has been shown to achieve very good performance at relatively low computational cost. Clear empirical evidence that training with residual connections accelerates the training of Inception networks significantly has been provided.

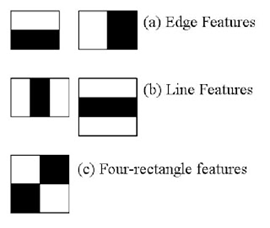
# ALGORITHMS USED

## Haar Cascades

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.

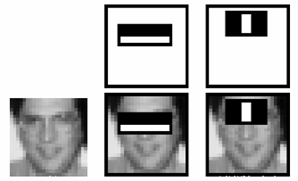
It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

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.



**Figure 10: Haar Like Features**

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.



**Figure 11: Using Haar Features**

We use the concept of adaboost which constructs a “strong” classifier as a linear combination of weighted simple “weak” classifiers.

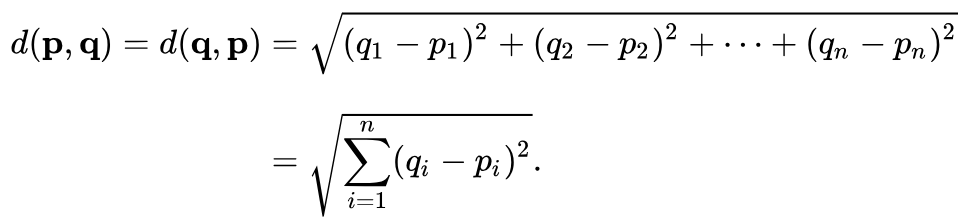
## Facenet

FaceNet was introduced in 2015 by Google researchers. It transforms the face into 128D Euclidean space similar to word embedding. Once the FaceNet model has been trained with triplet loss for different classes of faces to capture the similarities and differences between them, the 128-dimensional embedding returned by the FaceNet model can be used to cluster faces effectively. Once such a vector space (embedding) is created, tasks such as face recognition, verification and clustering can be easily implemented using standard techniques with FaceNet embeddings as feature vectors. In a way, distance would be closer for similar faces and further away for non-similar faces.

**Triplet Loss:**

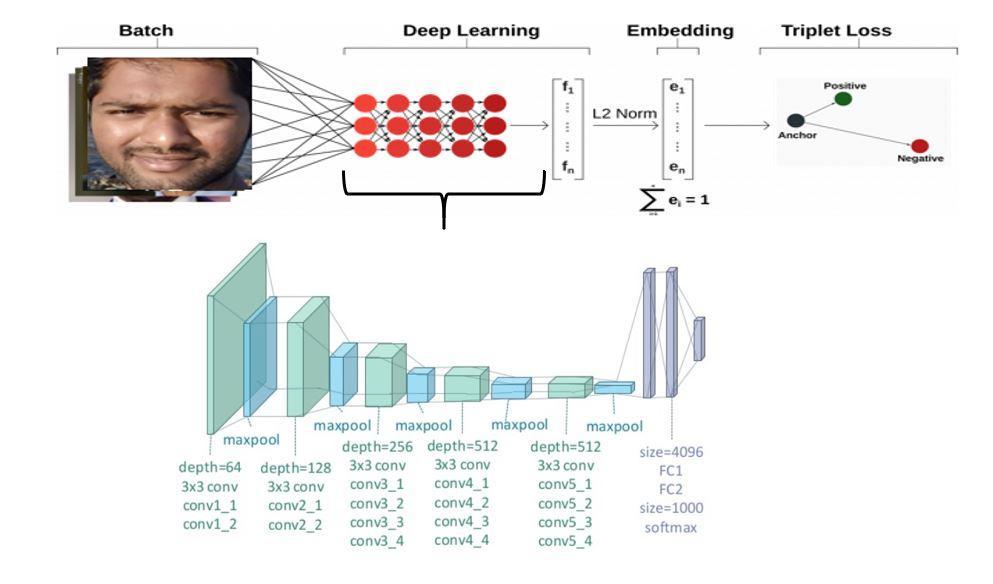
The Triplet Loss minimises the distance between an anchor and a positive, both of which have the same identity, and maximizes the distance between the anchor and a negative of a different identity.

Their practical differences lie in the difference of parameters and [FLOPS](https://en.wikipedia.org/wiki/FLOPS). The best model may be different depending on the application.



**Figure 12: Formula for finding the Euclidean distance between points p and q**

One the FaceNet model is trained, we can create the embedding for the face by feeding into the model. In order to compare two images, create the embedding for both images by feeding through the model separately. Then we can use the above formula to find the distance which will be lower value for similar faces and higher value for different faces.



**Figure 13: Training of a Facenet model**

**One shot learning using Facenet:**

One-shot learning is a classification task where one, or a few, examples are used to classify many new examples in the future.

This characterizes tasks as seen in the field of face recognition, such as face identification and face verification, where people must be classified correctly with different facial expressions, lighting conditions, accessories, and hairstyles given one or a few template photos.

Modern face recognition systems approach the problem of one-shot learning via face recognition by learning a rich low-dimensional feature representation, called a face embedding that can be calculated for faces easily and compared for verification and identification tasks.

Historically, embeddings were learned for one-shot learning problems using a Siamese network. The training of Siamese networks with comparative loss functions resulted in better performance, later leading to the triplet loss function used in the FaceNet system by Google that achieved then state-of-the-art results on benchmark face recognition tasks.

One-shot learning are classification tasks where many predictions are required given one (or a few) examples of each class, and face recognition is an example of one-shot learning.

Siamese networks are an approach to addressing one-shot learning in which a learned feature vector for the known and candidate example are compared.

Contrastive loss and later triplet loss functions can be used to learn high-quality face embedding vectors that provide the basis for modern face recognition systems.

# Our Solution

**Idea:**

In this project a face recognition based smart classroom attendance management system using computer vision and deep learning has been proposed. We propose that the student has to show his face to the camera while entering the classroom. We use face detection algorithms followed by face recognition and then mark the attendance of the detected student.

**Aim:**

* A decent accuracy of 90% in good lighting conditions. Low Cost Deployment. A Response time < 2Sec. Easy Installation and Maintenance Free.

**Working:**

We first capture the image of the student using the camera connected to the raspberry pi which is sent to the face detection model, the detected face is then passed to the deep learning model for further face recognition process. The results obtained are then updated in an excel sheet which can be sent to the corresponding department.

**Software and Algorithms used**:

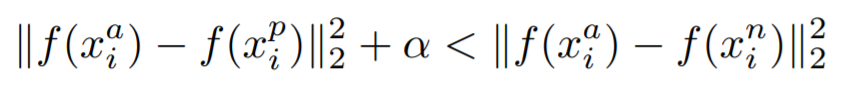
To implement face recognition, we used Python 3.6 along with OpenCV, Numpy, Pandas, Scikit-learn. The algorithm runs on the server with the Anaconda framework in the backend.

**Algorithm:**

We use Haar cascades algorithm to detect faces and Facenet algorithm to create the embedding for the detected faces. The embeddings obtained are then compared with the already generated embeddings during training using the euclidean distance. The Facenet is an algorithm developed by Google researchers which uses a specific architecture of convolutional neural networks and dense layers.

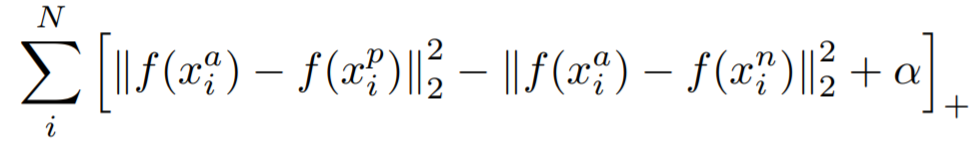
**Model Training:**

We train the neural network here to output a 128 dimensional vector for the input image. We strive for a network which outputs vectors which are close by for similar images and outputs a far away vector for dissimilar images. Thus we want



where alpha is the margin that we want between positive and negative pairs.

The loss function we therefore need to minimize is therefore given by

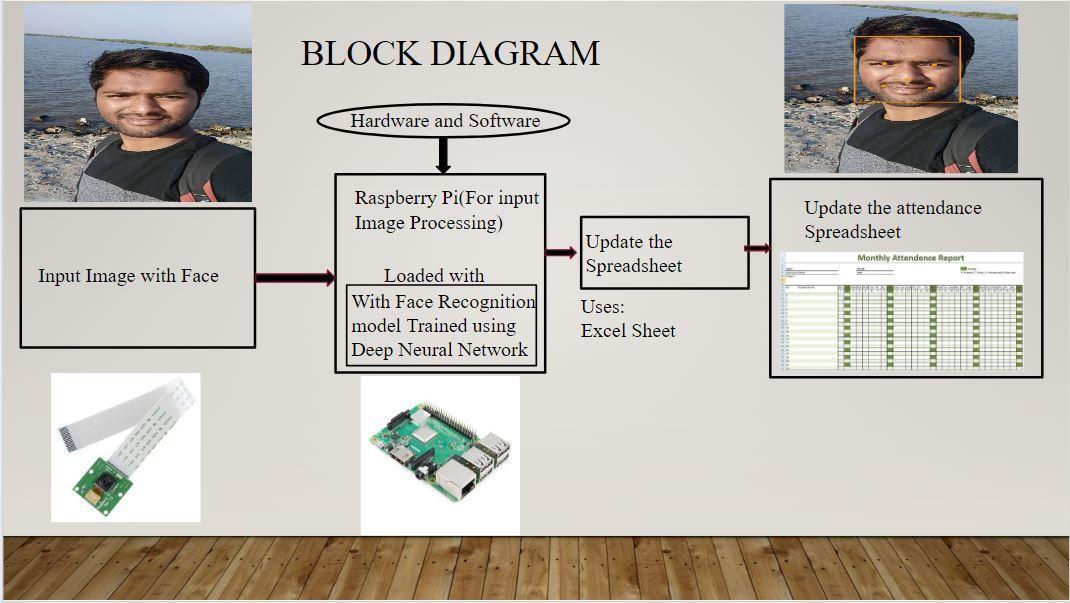


Generating all possible triplets would result in many triplets that easily satisfy the above equations. These triplets would not contribute to the training and result in slower convergence, as they would still be passed through the network. It is crucial to select hard triplets that are active and can therefore contribute to improving the model.

Training the facenet network to produce the correct embeddings can take days even with powerful GPUs so we use the pretrained networks open sourced by people at OpenFace. The good thing is once trained, the FaceNet network generates unique embeddings for faces on which it is not trained.

**Deployment:**

We deploy the algorithm in the Raspberry Pi connected to the Cameras through which we mark the attendance of the students in a CSV file.

****

**Figure 14: Prototype Deployment Block Diagram**

Our model is simple to construct and can be trained directly on full images. Once the network has been trained it scales well to any number of students provided that the memory and compute power is moderately available. It can also be made so that whenever an attendance is marked a message is sent to the student saying his attendance is marked.

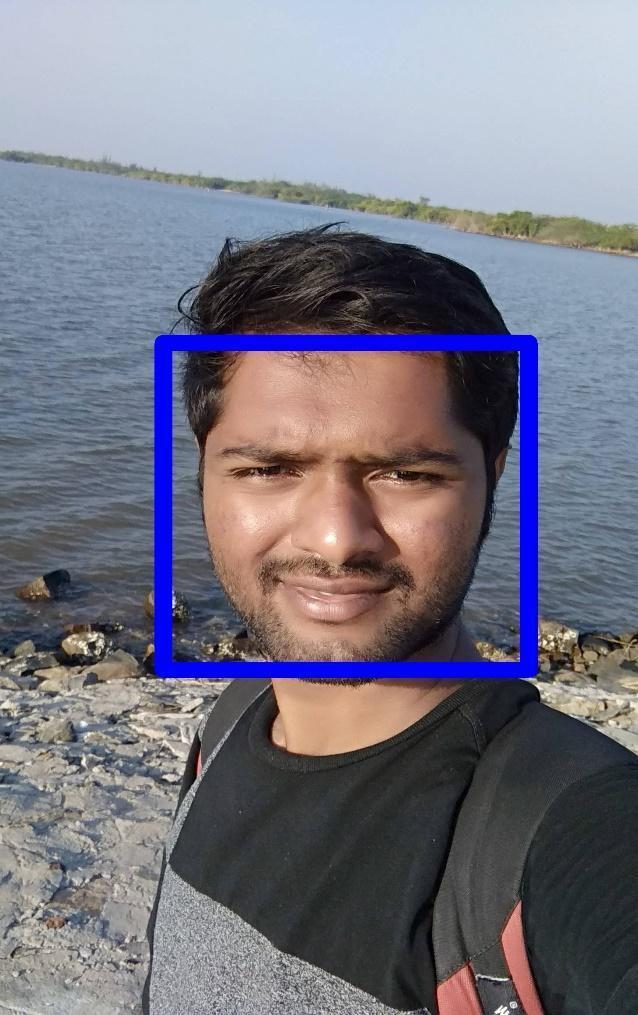
Thus, we can achieve a great level of accuracy in identifying and spotting out students' faces easily and accurately given more number of training photos.

# RESULTS

## FACE DETECTION USING HAAR CASCADES

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.

Initially we used MTCNN deep learning algorithm for detecting the faces but due to the low processing and power constraints of the Raspberry Pi it takes a lot of time just for detecting the face, We are therefore using Haar cascades for face detection as it is relatively computationally inexpensive.



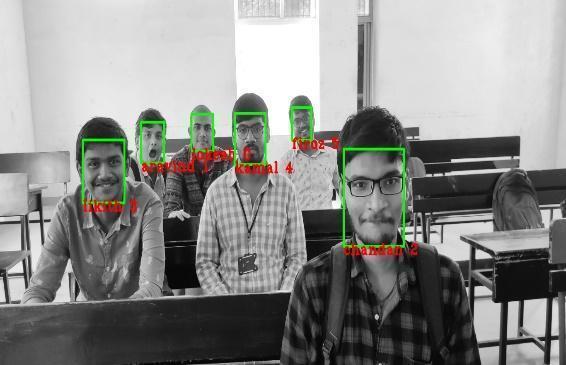
**Figure 15: Detected face using Haar cascades**

## FACE RECOGNITION USING FACENET

The facenet algorithm has been trained for 10 people with 30 images each and in that 6 people are made to sit in a classroom with ambient lighting on a fairly sunny day and when tested, all the students were correctly detected.

The processing time is around 2 seconds.





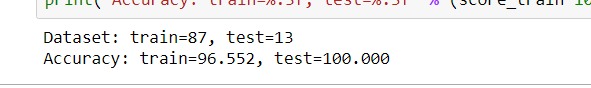
**Figure 16: Face recognition for multiple faces**

**Accuracy Measurement:**

The accuracy measurement is done by dividing the collected dataset in 90:10 ratio for the training and testing. So, the 90% of the data is fed for the model to get trained and the remaining 10% is taken a Test set to find the accuracy of the model.

To measure the accuracy the encodings generated by the model from the input images are collected as NumPy arrays and are then mapped to their respective known results followed by the Train and Test data split. Thus, the data which is obtained is said to have the minimum randomness to get fed to the model for accuracy measurement.

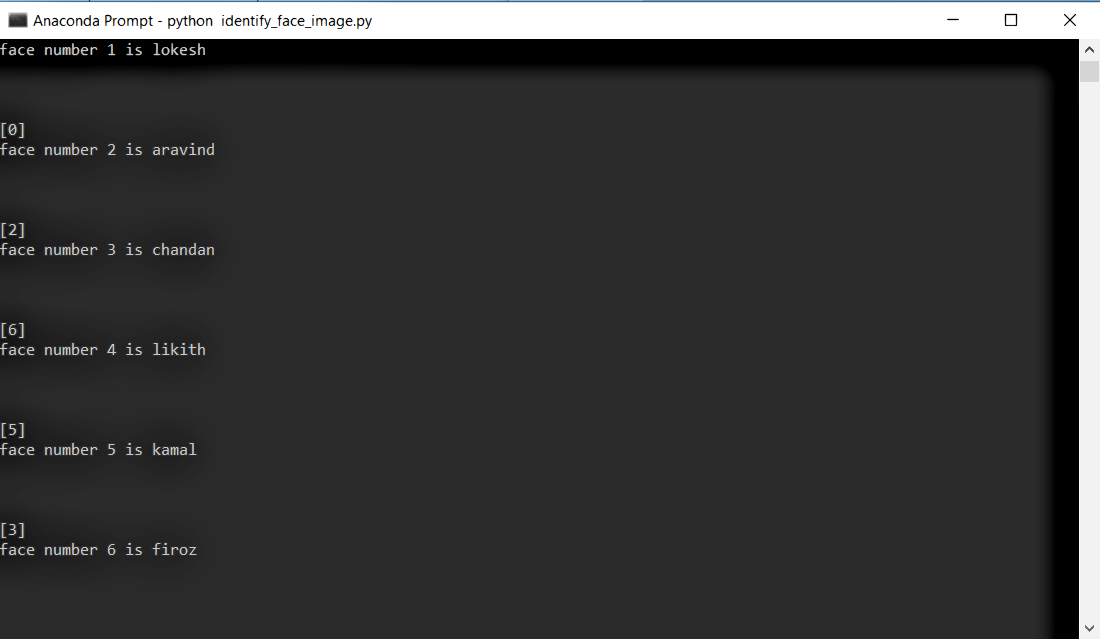
Using SVC(Support Vector Classifier ) model the accuracy is obtained as shown below.



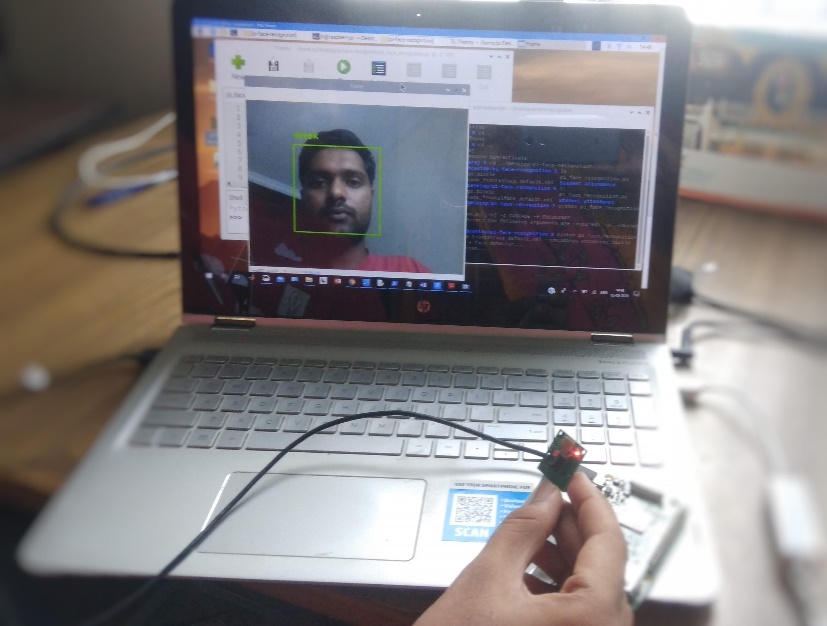
Accuracy Obtained with 87 training images and 13 test images. The model is so said to be 96.55% accurate and

The accuracy is affected by lighting conditions, training data quality and the poses of the individual faces and we are currently studying the effect of these parameters on the accuracy.

The below dialog box shows the terminal displaying the names of people along with their probabilities



**Figure 17: Command prompt displaying the names of students present**



**Figure 18: Raspberry pi recognising face.**

**Raspberry Pi is connected to the laptop using VNC viewer via ethernet**

# CONCLUSION

Considering the amount of time that is wasted in taking attendance and then analysing it, our solution saves an enormous amount of time in the long run of any academic institution. We implemented our face recognition system on the Raspberry pi which is more portable and requires less maintenance. The challenge faced for the used algorithm is that it requires high-resolution pictures of more than 30 images per each person

**FUTURE SCOPE:**

1. It has enormous potential in every office as it makes attendance much easier to track and prevent proxies.
2. It can be used in offices, schools and also by the security forces for identifying the criminals.
3. It can be implemented on mobile phones using APIs and therefore attendance can be taken using the mobile of the professor.

# Bibliography

1. A. Mittal, F. S. Khan, P. Kumar and T. Choudhury,. “Cloud based intelligent attendance system through video streaming,.” *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon), Bangalore, 2017,*. 2017.
2. Alex Krizhevsky, Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional.” *Magazine Communications of the ACM* (2012).
3. Borra, N. K. Jayant and S. “Attendance management system using hybrid face recognition techniques.” *Conference on Advances in Signal Processing (CASP)*. 2016.
4. Dika, V. Shehu and A. “Using real time computer vision algorithms in automatic attendance management systems.” *Proceedings of the ITI 2010, 32nd International Conference on Information Technology Interfaces, Cavtat,*. 2010.
5. “ImageNet Classification with Deep Convolutional.” *Alex Krizhevsky,Geoffrey E. Hinton* (2012).
6. J. Edwin, M. Greeshma, M. T. P. Haridas and M. H. Supriya. “Face Recognition based Surveillance System Using FaceNet and MTCNN on Jetson TX2.” *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*. n.d.
7. K. Puthea, R. Hartanto and R. Hidayat. “A review paper on attendance marking system based on face recognition,.” 2017: 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE), n.d.
8. S. K. Gupta, A. T S and R. M. Reddy Guddeti. “CVUCAMS: Computer Vision Based Unobtrusive Classroom Attendance Management System,.” *2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT),*. 2018.
9. Schroff, Florian, Dmitry Kalenichenko, and James Philbin. “FaceNet: A Unified Embedding for Face Recognition and Clustering.” *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)*. 2015.
10. Tanriverdi, R. Samet and M. “Face Recognition-Based Mobile Automatic Classroom Attendance Management System.” *2017 International Conference on Cyberworlds* . 2017.
11. Zhang, Kaipeng and Zhang, Zhanpeng and Li, Zhifeng and Qiao, Yu. *IEEE Signal Processing Letters (2016)*. Institute of Electrical and Electronics Engineers (IEEE)}, 2016. http://dx.doi.org/10.1109/LSP.2016.2603342.