Visualizing Website Clickstream Data

*A Course Project Report submitted to the*

# JAWAHARLAL NEHRU TECHNOLOGICALUNIVERSITY HYDERABAD

*in partial fulfillment of the requirements for the award of the degree of*

# BACHELOR OF TECHNOLOGY

IN

# COMPUTER SCIENCE & ENGINEERING

**Submitted by**

**D SIDDHARTH 20071A6712**

**D NEERAJ ASHISH 20071A6713**



**VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY**

An Autonomous Institute, NAAC Accredited with ‘A++’ Grade (CGPA: 3.73/4.0) NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B.Tech. Programmes

Approved by AICTE, New Delhi, Affiliated to JNTU-H, Recognised as “College with Potential for Excellence” by UGC VignanaJyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad TS 500 090 India

**2021**

## VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with ‘A++’ Grade (CGPA: 3.73/4.0) NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B.Tech. Programmes

Approved by AICTE, New Delhi, Affiliated to JNTU-H, Recognised as “College with Potential for Excellence” by UGC VignanaJyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad TS 500 090 India



**CERTIFICATE**

This is to certify that D. Siddharth **(20071A6712)**, **D Neeraj Ashish(20071A6713** have successfully completed their Course Based Project work at **Computer Science & Engineering Department** of **Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology,** Hyderabad entitled **“FACE DETECTION USING PYTHON AND OPEN CV”** in partial fulfillment of the requirements for the award of **B.Tech** during the academic year 2021-2022**.**

This work is carried out under my supervision and has not been submitted to any other University/ Institute for award of any degree/ diploma.

|  |  |  |
| --- | --- | --- |
| **Mr. Sunil Kumar Assistant Professor**  **CSE Department** |  | **Dr. Rajashekar**  **Associate Professor and HoD CSE Department** |
| **VNRVJIET** |  | **VNRVJIET** |
| **Hyderabad** |  | **Hyderabad** |
|  | **External Examiners** |  |

DECLARATION

This is to certify that the project work entitled **“FACE DETECTION USING PYTHON AND OPEN CV”** submitted in VNR Vignana Jyothi Institute of Engineering & Technology in partial fulfillment of requirement for the award of Bachelor of Technology in Computer Science and Engineering is a bonafide report of the work carried out by us under the guidance and supervision of Dr. Kranthi kumar, Assistant Professor, Department of CSEDS, VNRVJIET. To the best of our knowledge, this report has not been submitted in any form to any university or institution for the award of any degree or diploma.

Submission Dt:

|  |  |  |  |
| --- | --- | --- | --- |
| **D.SIddharth** | **D. Neeraj Ashish** |  |  |
| (20071A6712) | (20071A6713) |  |  |
| II B.Tech-CSE, | II B.Tech-CSE, |  |  |
| VNR VJIET | VNR VJIET |  |  |

# ACKNOWLEDGEMENT

An endeavor over a long period can be successful only with the advice and support of many well-wishers. We take this opportunity to express our gratitude and appreciation to all of them.

First of all we thank the lord almighty who has been with us from the beginning to the end of our project. We are indebted to our venerable principal **Dr. C. D. Naidu** for this unflinching devotion, which led us to complete this project. The support, encouragement given by him and his motivation lead us to complete this project.

We wish to express our profound gratitude to **Dr. Rajashekar, Associate Professor** and **HOD CSE DS** Department**, VNR Vignana Jyothi Institute of Engineering and Technology** for their constant and dedicated service to brighten our career.

With great pleasure we express our gratitude to the internal guide **Dr. M Kranthi kumar, Assistant Professor, CSE** for his timely help, constant guidance, cooperation, support and encouragement throughout this project.

Finally we wish to express our deep sense of gratitude and sincere thanks to our parents, friends and all our well-wishers who have technically and non-technically contributed for the successful completion of our course based project.

D. SIDDHARTH (20071A6712)

D. NEERAJ ASHISH (20071A6713)

# FACE DETECTION SYSTEM

## ABSTRACT

The face is one of the easiest ways to distinguish the individual identity of each other. Face recognition is a personal identification system that uses personal characteristics of a person to identify the person's identity. Human face recognition procedure basically consists of two phases, namely face detection, where this process takes place very rapidly in humans, except under conditions where the object is located at a short distance away, the next is the introduction, which recognize a face as individuals. Stage is then replicated and developed as a model for facial image recognition (face recognition) is one of the much-studied biometrics technology and developed by experts. There are two kinds of methods that are currently popular in developed face recognition pattern namely, Eigenface method and Fisherface method. Facial image recognition Eigenface method is based on the reduction of face- dimensional space using Principal Component Analysis (PCA) for facial features. The main purpose of the use of PCA on face recognition using Eigen faces was formed (face space) by finding the eigenvector corresponding to the largest eigenvalue of the face image. The area of this project face detection system with face recognition is Image processing. The software requirements for this project is matlab software.

Keywords: face detection, Eigen face, PCA, matlab

Extension: There are vast number of applications from this face detection project, this project can be extended that the various parts in the face can be detect which are in various directions and shapes.

# INDEX

**CONTENT PAGE**

[CHAPTER 1: INTRODUCTION](https://docs.google.com/document/d/1kjPdhwTKK_REmpJAqRvLJKG986yL7SxS4TGm2EV-qrQ/edit#heading%3Dh.qj93bymhj6ot) 1

* 1. Introduction 1
  2. Application Overview 3
  3. Introduction to Technologies used 4
     1. Hive 4

[CHAPTER 2: LITERATURE SURVEY](https://docs.google.com/document/d/1kjPdhwTKK_REmpJAqRvLJKG986yL7SxS4TGm2EV-qrQ/edit#heading%3Dh.uz77fj9q9jyx) 7

[CHAPTER 3](https://docs.google.com/document/d/1kjPdhwTKK_REmpJAqRvLJKG986yL7SxS4TGm2EV-qrQ/edit#heading%3Dh.tbioi9yyhox3) SOFTWARE REQUIREMENT ANALYSIS 10

* 1. Functional Requirements 10
  2. Nonfunctional Requirements 11

[CHAPTER 4:](https://docs.google.com/document/d/1kjPdhwTKK_REmpJAqRvLJKG986yL7SxS4TGm2EV-qrQ/edit#heading%3Dh.flk7uvb28qxi) SOFTWARE DESIGN 12

* 1. [UML Diagrams](https://docs.google.com/document/d/1kjPdhwTKK_REmpJAqRvLJKG986yL7SxS4TGm2EV-qrQ/edit#heading%3Dh.2p2csry) 12
     1. [Use Case Diagram](https://docs.google.com/document/d/1kjPdhwTKK_REmpJAqRvLJKG986yL7SxS4TGm2EV-qrQ/edit#heading%3Dh.147n2zr) 14
     2. Activity [Diagram](https://docs.google.com/document/d/1kjPdhwTKK_REmpJAqRvLJKG986yL7SxS4TGm2EV-qrQ/edit#heading%3Dh.nv6j2khqug9n) 16

[CHAPTER 5](https://docs.google.com/document/d/1kjPdhwTKK_REmpJAqRvLJKG986yL7SxS4TGm2EV-qrQ/edit#heading%3Dh.b8psjyx1lk1w) : IMPLEMENTATION 20

[CHAPTER 6](https://docs.google.com/document/d/1kjPdhwTKK_REmpJAqRvLJKG986yL7SxS4TGm2EV-qrQ/edit#heading%3Dh.b8psjyx1lk1w) : TESTING 34

* 1. Testing Plan 36
  2. White Box Testing 36
     1. Model Testing vi 36

# CHAPTER-1 INTRODUCTION

Face recognition is the task of identifying an already detected object as a known or unknown face.Often the problem of face recognition is confused with the problem of face detectionFace Recognition on the other hand is to decide if the "face" is someone known, or unknown, using for this purpose a database of faces in order to validate this input face.

## FACE RECOGNIZATION:

DIFFERENT APPROACHES OF FACE RECOGNITION:

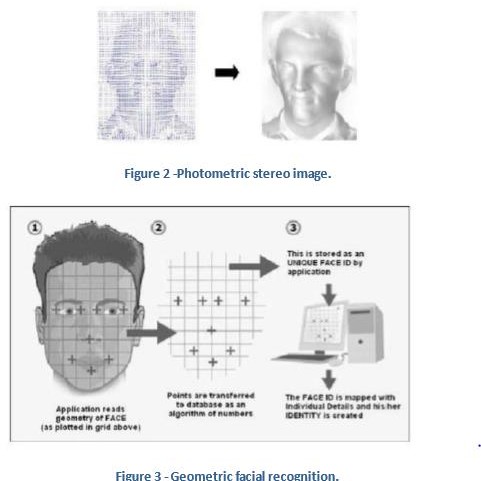
There are two predominant approaches to the face recognition problem: Geometric (feature based) and photometric (view based). As researcher interest in face recognition continued, many different algorithms were developed, three of which have been well studied in face recognition literature.

### Recognition algorithms can be divided into two main approaches:

* + 1. **Geometric:** Is based on geometrical relationship between facial landmarks, or in other words the spatial configuration of facial features. That means that the main geometrical features of the face such as the eyes, nose and mouth are first located and then faces are classified on the basis of various geometrical distances and angles between features. (Figure 3)
    2. **Photometric stereo:** Used to recover the shape of an object from a number of images taken under different lighting conditions. The shape of the recovered object is defined by a gradient map, which is made up of an array of surface normals (Zhao and Chellappa, 2006) (Figure 2)

### Popular recognition algorithms include:

* + - 1. Principal Component Analysis using Eigenfaces, (PCA)
      2. Linear Discriminate Analysis,
      3. Elastic Bunch Graph Matching using the Fisherface algorithm,



.

## FACE DETECTION:

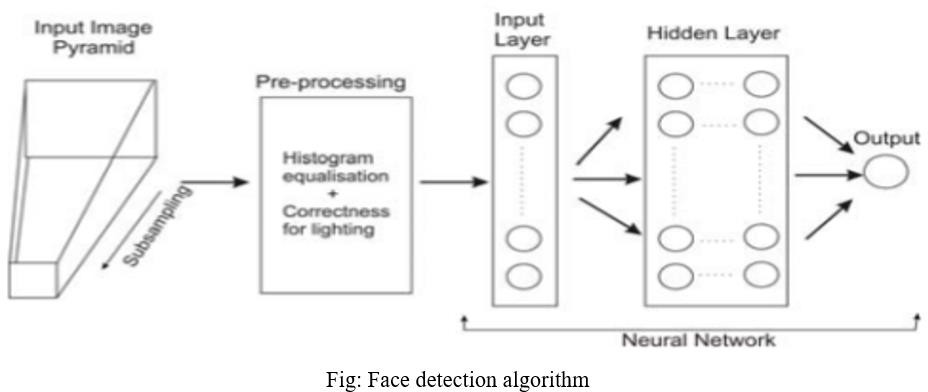
Face detection involves separating image windows into two classes; one containing faces (tarning the background (clutter). It is difficult because although commonalities exist between faces, they can vary considerably in terms of age, skin colour and facial expression. The problem is further complicated by differing lighting conditions, image qualities and geometries, as well as the possibility of partial occlusion and disguise. An ideal face detector would therefore be able to detect the presence of any face under any set of lighting conditions, upon any background. The face detection task can be broken down into two steps. The first step is a classification task that takes some arbitrary image as input and outputs a binary value of yes or no, indicating whether there are any faces present in the image. The second step is the face localization task that aims to take an image as input and output the location of any face or faces within that image as some bounding box with (x, y, width, height).

### The face detection system can be divided into the following steps:-

1. **Pre-Processing:** To reduce the variability in the faces, the images are processed before they are fed into the network. All positive examples that is the face images are obtained by cropping

images with frontal faces to include only the front view. All the cropped images are then corrected for lighting through standard algorithms.

1. **Classification:** Neural networks are implemented to classify the images as faces or nonfaces by training on these examples. We use both our implementation of the neural network and the Matlab neural network toolbox for this task. Different network configurations are experimented with to optimize the results.
2. **Localization:** The trained neural network is then used to search for faces in an image and if present localize them in a bounding box. Various Feature of Face on which the work has done on:- Position Scale Orientation Illumination



# CHAPTER 1

Face recognition is the task of identifying an already detected object as a known or unknown face.Often the problem of face recognition is confused with the problem of face detectionFace Recognition on the other hand is to decide if the "face" is someone known, or unknown, using for this purpose a database of faces in order to validate this input face.

## FACE RECOGNIZATION:

DIFFERENT APPROACHES OF FACE RECOGNITION:

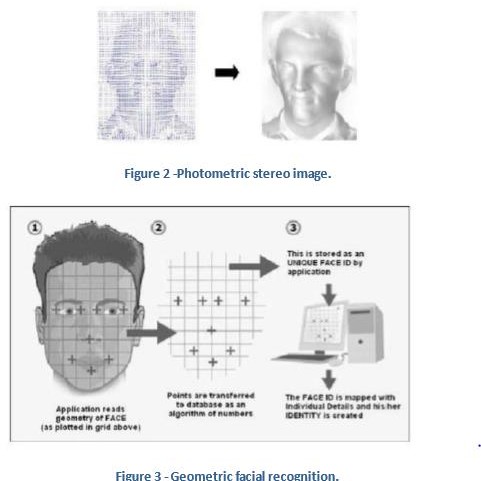
There are two predominant approaches to the face recognition problem: Geometric (feature based) and photometric (view based). As researcher interest in face recognition continued, many different algorithms were developed, three of which have been well studied in face recognition literature.

### Recognition algorithms can be divided into two main approaches:

* + 1. **Geometric:** Is based on geometrical relationship between facial landmarks, or in other words the spatial configuration of facial features. That means that the main geometrical features of the face such as the eyes, nose and mouth are first located and then faces are classified on the basis of various geometrical distances and angles between features. (Figure 3)
    2. **Photometric stereo:** Used to recover the shape of an object from a number of images taken under different lighting conditions. The shape of the recovered object is defined by a gradient map, which is made up of an array of surface normals (Zhao and Chellappa, 2006) (Figure 2)

### Popular recognition algorithms include:

* + - 1. Principal Component Analysis using Eigenfaces, (PCA)
      2. Linear Discriminate Analysis,
      3. Elastic Bunch Graph Matching using the Fisherface algorithm,



.

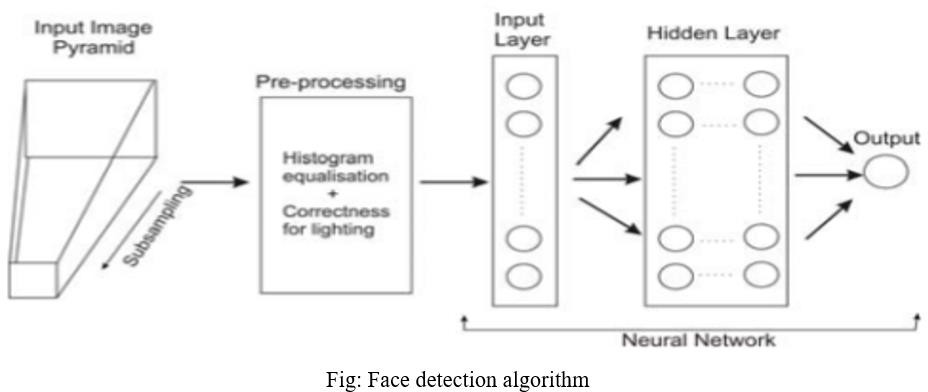
## FACE DETECTION:

Face detection involves separating image windows into two classes; one containing faces (tarning the background (clutter). It is difficult because although commonalities exist between faces, they can vary considerably in terms of age, skin colour and facial expression. The problem is further complicated by differing lighting conditions, image qualities and geometries, as well as the possibility of partial occlusion and disguise. An ideal face detector would therefore be able to detect the presence of any face under any set of lighting conditions, upon any background. The face detection task can be broken down into two steps. The first step is a classification task that takes some arbitrary image as input and outputs a binary value of yes or no, indicating whether there are any faces present in the image. The second step is the face localization task that aims to take an image as input and output the location of any face or faces within that image as some bounding box with (x, y, width, height).

### The face detection system can be divided into the following steps:-

**Pre-Processing:** To reduce the variability in the faces, the images are processed before they are fed into the network. All positive examples that is the face images are obtained by cropping images with frontal faces to include only the front view. All the cropped images are then corrected for lighting through standard algorithms.

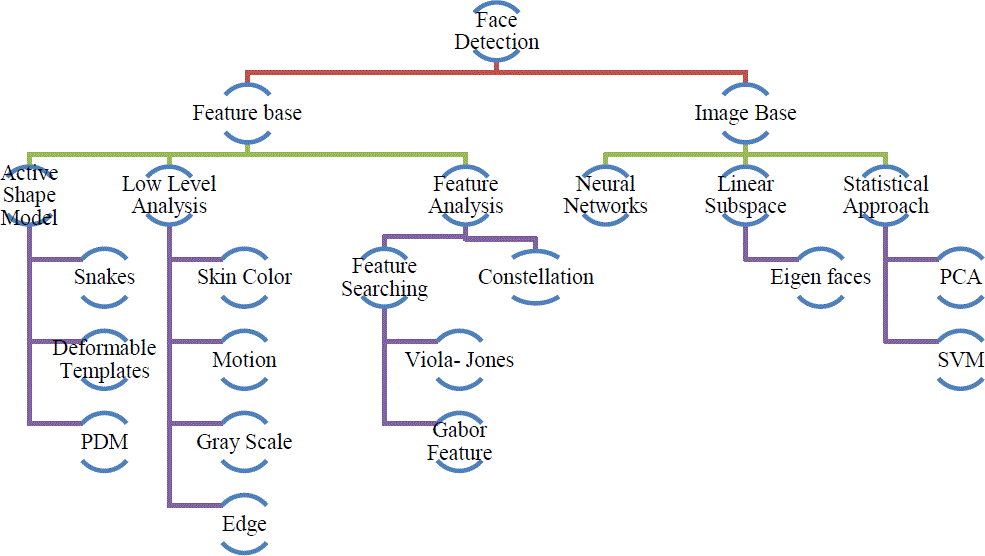
1. **Classification:** Neural networks are implemented to classify the images as faces or nonfaces by training on these examples. We use both our implementation of the neural network and the Matlab neural network toolbox for this task. Different network configurations are experimented with to optimize the results.
2. **Localization:** The trained neural network is then used to search for faces in an image and if present localize them in a bounding box. Various Feature of Face on which the work has done on:- Position Scale Orientation Illumination



# CHAPTER – 2:

# LITERATURE SURVEY

Face detection is a computer technology that determines the location and size of human face in arbitrary (digital) image. The facial features are detected and any other objects like trees, buildings and bodies etc are ignored from the digital image. It can be regarded as a ‗specific‘ case of object-class detection, where the task is finding the location and sizes of all objects in an image that belong to a given class. Face detection, can be regarded as a more ‗general‘ case of face localization. In face localization, the task is to find the locations and sizes of a known number of faces (usually one). Basically there are two types of approaches to detect facial part in the given image i.e. feature base and image base approach.Feature base approach tries to extract features of the image and match it against the knowledge of the face features. While image base approach tries to get best match between training and testing images.



**Fig 2.1 detection methods**

## 

## FEATURE BASE APPROCH:

Active Shape ModelActive shape models focus on complex non-rigid features like actual physical and higher level appearance of features Means that Active Shape Models (ASMs) are aimed at automatically locating landmark points that define the shape of any statistically modelled

object in an image. When of facial features such as the eyes, lips, nose, mouth and eyebrows. The training stage of an ASM involves the building of a statistical

* + 1. facial model from a training set containing images with manually annotated landmarks.

ASMs is classified into three groups i.e. snakes, PDM, Deformable templates

* + 1. 1.1)Snakes:The first type uses a generic active contour called snakes, first introduced by Kass et al. in 1987 Snakes are used to identify head boundaries [8,9,10,11,12]. In order to achieve the task, a snake is first initialized at the proximity around a head boundary. It then locks onto nearby edges and subsequently assume the shape of the head. The evolution of a snake is achieved by minimizing an energy function, Esnake (analogy with physical systems), denoted asEsnake = Einternal + EExternal WhereEinternal and EExternal are internal and external energy functions.Internal energy is the part that depends on the intrinsic properties of the snake and defines its natural evolution. The typical natural evolution in snakes is shrinking or expanding. The external energy counteracts the internal energy and enables the contours to deviate from the natural evolution and eventually assume the shape of nearby features—the head boundary at a state of equilibria.Two main consideration for forming snakes i.e. selection of energy terms and energy minimization. Elastic energy is used commonly as internal energy. Internal energy is vary with the distance between control points on the snake, through which we get contour an elastic-band characteristic that causes it to shrink or expand. On other side external energy relay on image features. Energy minimization process is done by optimization techniques such as the steepest gradient descent. Which needs highest computations. Huang and Chen and Lam and Yan both employ fast iteration methods by greedy algorithms.

# CHAPTER 3

### SOFTWARE REQUIREMENT ANALYSIS

### FUNCTIONAL REQUIREMENTS :

## DIGITAL IMAGE PROCESSING

### Interest in digital image processing methods stems from two principal application areas:

* + 1. Improvement of pictorial information for human interpretation
    2. Processing of scene data for autonomous machine perception

In this second application area, interest focuses on procedures for extracting image information in a form suitable for computer processing.

Examples includes automatic character recognition, industrial machine vision for product assembly and inspection, military recognizance, automatic processing of fingerprints etc.

### Image:

Am image refers a 2D light intensity function f(x, y), where(x, y) denotes spatial coordinates and the value of f at any point (x, y) is proportional to the brightness or gray levels of the image at that point. A digital image is an image f(x, y) that has been discretized both in spatial coordinates and brightness. The elements of such a digital array are called image elements or pixels.

### A simple image model:

To be suitable for computer processing, an image f(x, y) must be digitalized both spatially and in amplitude. Digitization of the spatial coordinates (x, y) is called image sampling. Amplitude digitization is called gray-level quantization.

The storage and processing requirements increase rapidly with the spatial resolution and the number of gray levels.

Example: A 256 gray-level image of size 256x256 occupies 64k bytes of memory.

### Types of image processing

* Low level processing
* Medium level processing
* High level processing

Low level processing means performing basic qperations on images such as reading an image resize, resize, image rotate, RGB to gray level conversion, histogram equalization etc…, The output image obtained after low level processing is raw image.Medium level processing means extracting regions of interest from output of low level processed image. Medium level processing deals with identification of boundaries i.e edges .This process is called segmentation.High level processing deals with adding of artificial intelligence to medium level processed signal.

## FUNDAMENTAL STEPS IN IMAGE PROCESSING

### Fundamental steps in image processing are

* + 1. Image acquisition: to acquire a digital image
    2. Image pre-processing: to improve the image in ways that increases the chances for success of the other processes.
    3. Image segmentation: to partitions an input image into its constituent parts of objects.
    4. Image segmentation: to convert the input data to a from suitable for computer processing.
    5. Image description: to extract the features that result in some quantitative information of interest of features that are basic for differentiating one class of objects from another.
    6. Image recognition: to assign a label to an object based on the information provided by its Description.
    7. Image recognition: to assign a label to an object based on the information provided by its description.

Pre-processing

Segmentation

Recognition And

Representation and description

Knowledge base



problem

fig.3.1. Fundamental steps in digital image processing

## ELEMENTS OF DIGITAL IMAGE PROCESSING SYSTEMS

A digital image processing system contains the following blocks as shown in the figure

Processing Unit

* Computer
* Work station

Display unit

* TV monitors
* Printers
* Projectors

Image acquisition equipments

* Video
* Scanner

Storage

* Optical discs
* Tape
* Video tape
* Magnetic discs

Communication channel

Fig.3.3. Elements of digital image processing systems

The basic operations performed in a digital image processing system include

* + 1. Acquisition
    2. Storage
    3. Processing
    4. Communication
    5. Display

## A simple image formation model

Image are denoted by two-dimensional function f(x, y).f(x, y) may be characterized by 2 components:

* + - 1. The amount of source illumination i(x, y) incident on the scene
      2. The amount of illumination reflected r(x, y) by the objects of the scene 3. f(x, y) = i(x, y)r(x, y), where 0 < i(x,y) < and 0 < r(x, y) < 1

### Typical values of reflectance r(x, y):

* + - * + 0.01 for black velvet
        + 0.65 for stainless steel
        + 0.8 for flat white wall paint
        + 0.9 for silver-plated metal
        + 0.93 for snow Example of typical ranges of illumination i(x, y) for visible light (average values)
* Sun on a clear day: ~90,000 lm/m^2,down to 10,000lm/m^2 on a cloudy day
* Full moon on a clear evening :-0.1 lm/m^2
* Typical illumination level in a commercial office. ~1000lm/m^2 image Formats (supported by MATLAB Image Processing Toolbox)

|  |  |  |  |
| --- | --- | --- | --- |
| Format  name | Full name | Description | Recognized  extensions |
| TIFF | Tagged Image File Format | A flexible file format supporting a variety image compression standards including  JPEG | .tif, .tiff |
| JPEG | Joint Photographic Experts Group | A standard for  compression of images of photographic quality | .jpg, .jpeg |
| GIF | Graphics Interchange Format | Frequently used to make small animations  on the internet | .gif |
| BMP | Windows Bitmap | Format used mainly for simple uncompressed  images | .bmp |
| PNG | Portable Network Graphics | Compresses full color images with  trasparency(up to  48bits/p | .png |

Table.3.3. Image Formats Supported By MATLAB

# CHAPTER 4

### SOFTWARE DESIGN

### UML Diagrams

Unified Modeling Language is a tool that helps a designer to present his ideas about the project to his client and his developer. Modeling plays a crucial role in designing software. A poorly designed model can lead to poorly developed software.

A UML system has five different views that help in describing systems from different perspectives. Each view has a set of diagrams and components that represent the real time objects.

### User Model View

* + It models the user behavior in a system context.
  + All the diagrams are drawn keeping in mind the user’s response and reaction towards a system.

### Structural Model View

* + This view consists of a class diagram and object diagram which is used to model the static structures.
  + It uses objects, attributes, operations and relationships.

### Behavioral Model View

* + It mainly consists of the sequence diagram, collaboration diagram, state chart diagram and activity diagram. They mainly represent flow of actions between different objects involved in the system
  + They are used to visualize various dynamic aspects of the system architecture.

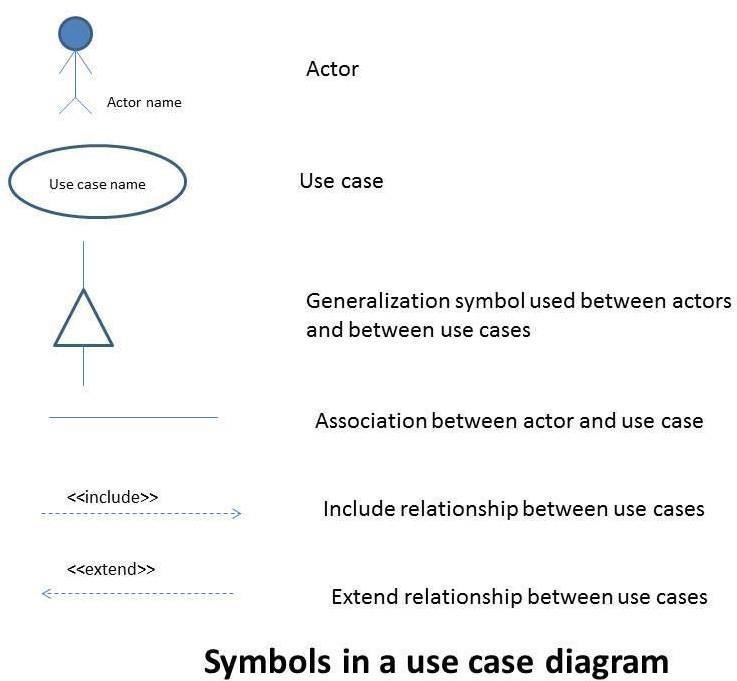
### Implementation Model View

* + This view consists of component diagrams and deployment diagrams. This view models the static software modules for an organization.
  + This usually contains the data files, documentation, the executables and source code
  + These are the physically replaceable components of the system. They are modeled using component diagrams

### Use Case Diagram

The basic representation for the interaction of the user with the system is represented using the use case diagram. It involves the relationship between the user and various use cases with the actors being involved. There are different kinds of relationships that are involved between the use cases and the actors. They include:

1. Association relationship
2. Generalization
3. Dependency
4. Realizations
5. Transitions



**The following image represents the Use Case diagram of the proposed system:** In the Use case diagram, Web Customer [actor](https://www.uml-diagrams.org/use-case-actor.html) uses some web site to make purchases online. Top level [use cases](https://www.uml-diagrams.org/use-case.html) are View Items, Make Purchase and Client Register. View Items use case could be used by customer as top level use case if customer only wants to find and see some products. This use case could also be used as a part of Make Purchase use case. Client Register use case allows customer to register on the web site, for example to get some coupons or be invited to private sales. Note, that Checkout use case is [included use case](https://www.uml-diagrams.org/use-case-include.html) not available by itself - checkout is part of making purchase.

### 

### Activity Diagram

The flow from one activity to another activity can be represented in the form of a flow chart which is usually an activity diagram. It forms a backbone for the UML diagrams. It depicts the dynamic aspects for all the objects within the system.

The control flow from one object to another object is drawn which shows the basic operations that are to be performed.

The basic purpose of an activity diagram is the same as that of other UML diagrams. The dynamic behavior of the system is viewed by the activity diagram. They are used to construct a system using the backward and forward engineering mechanisms.

The purpose of an activity diagram is as follows:

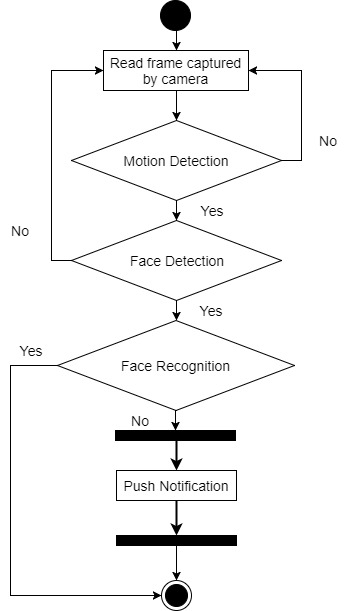
* + - 1. For drawing the flow (i.e. activity) in a system.
      2. For showing the flow of sequence from one activity to another activity.
      3. For showing the concurrent and parallel flow of actions in the system

The elements that are used in an activity diagram are as follows:

* + - * 1. Association relationship
        2. Activities
        3. Conditions and Constraints

The following represents the Activity diagram of the proposed system:

Firstly we need to use a webcam to capture images of faces which are used for face detection. Prepare a dataset of large images so that our model is accurate enough and split into training and testing images. Now we need to extract features like eigenfaces from these training images which are then used to train with the convolutional neural network. When capturing frames if motion is captured then check face detection. If the face is detected, then it tries to recognize your face.



# CHAPTER 5

**Implementation**

## About Face detection

Face detection is a computer vision problem that involves finding faces in photos.

It is a trivial problem for humans to solve and has been solved reasonably well by classical feature-based techniques, such as the cascade classifier. More recently deep learning methods have achieved state-of-the-art results on standard benchmark face detection datasets. One example is the Multi-task Cascade Convolutional Neural Network, or MTCNN for short.

Locating a face in a photograph refers to finding the coordinate of the face in the image, whereas localization refers to demarcating the extent of the face, often via a bounding box around the face.

## Test Photographs

We need test images for face detection in this tutorial.

To keep things simple, we will use two test images: one with two faces, and one with many faces. We’re not trying to push the limits of face detection, just demonstrate how to perform face detection with normal front-on photographs of people.

# 

# 

## Face Detection With OpenCV

Feature-based face detection algorithms are fast and effective and have been used successfully for decades.

Perhaps the most successful example is a technique called cascade classifiers first described by Paul Viola and Michael Jones and their 2001 paper titled “[Rapid Object Detection using a Boosted Cascade of Simple Features](https://ieeexplore.ieee.org/document/990517).”

In the paper, effective features are learned using the [AdaBoost algorithm](https://machinelearningmastery.com/boosting-and-adaboost-for-machine-learning/), although importantly, multiple models are organized into a hierarchy or “cascade.”

In the paper, the AdaBoost model is used to learn a range of very simple or weak features in each face, that together provide a robust classifier.

*… feature selection is achieved through a simple modification of the AdaBoost procedure: the weak learner is constrained so that each weak classifier returned can depend on only a single feature . As a result each stage of the boosting process, which selects a new weak classifier, can be viewed as a feature selection process.*

— [Rapid Object Detection using a Boosted Cascade of Simple Features](https://ieeexplore.ieee.org/document/990517), 2001.

The models are then organized into a hierarchy of increasing complexity, called a “cascade“.

Simpler classifiers operate on candidate face regions directly, acting like a coarse filter, whereas complex classifiers operate only on those candidate regions that show the most promise as faces.

*… a method for combining successively more complex classifiers in a cascade structure which dramatically increases the speed of the detector by focusing attention on promising regions of the image.*

— [Rapid Object Detection using a Boosted Cascade of Simple Features](https://ieeexplore.ieee.org/document/990517), 2001.

The result is a very fast and effective face detection algorithm that has been the basis for face detection in consumer products, such as cameras.

*Their detector, called detector cascade, consists of a sequence of simple-to-complex face classifiers and has attracted extensive research efforts. Moreover, detector cascade has been deployed in many commercial products such as smartphones and digital cameras.*

— [Multi-view Face Detection Using Deep Convolutional Neural Networks](https://arxiv.org/abs/1502.02766), 2015.

It is a modestly complex classifier that has also been tweaked and refined over the last nearly 20 years.

A modern implementation of the [Classifier Cascade](https://en.wikipedia.org/wiki/Cascading_classifiers) face detection algorithm is provided in the [OpenCV library](https://opencv.org/). This is a C++ computer vision library that provides a python interface. The benefit of this implementation is that it provides pre-trained face detection models, and provides an interface to train a model on your own dataset.

OpenCV can be installed by the package manager system on your platform, or via pip; for example:



Once the installation process is complete, it is important to confirm that the library was installed correctly.

This can be achieved by importing the library and checking the version number; for example:



OpenCV provides the [CascadeClassifier class](https://docs.opencv.org/3.4.3/d1/de5/classcv_1_1CascadeClassifier.html) that can be used to create a cascade classifier for face detection. The constructor can take a filename as an argument that specifies the XML file for a pre-trained model.

OpenCV provides a number of pre-trained models as part of the installation. These are available on your system and are also available on the [OpenCV GitHub project](https://github.com/opencv/opencv/tree/master/data/haarcascades).

Download a pre-trained model for frontal face detection from the OpenCV GitHub project and place it in your current working directory with the filename ‘haarcascade\_frontalface\_default.xml‘.

Once downloaded, we can load the model as follows:



Once loaded, the model can be used to perform face detection on a photograph by calling the [detectMultiScale() function](https://docs.opencv.org/3.4.3/d1/de5/classcv_1_1CascadeClassifier.html" \l "aaf8181cb63968136476ec4204ffca498).

This function will return a list of bounding boxes for all faces detected in the photograph.



We can demonstrate this with an example with the college students photograph (*test.jpg*).

The photo can be loaded using OpenCV via the *imread()* function.



The complete example of performing face detection on the college students photograph with a pre-trained cascade classifier in OpenCV is listed below.



Running the example first loads the photograph, then loads and configures the cascade classifier; faces are detected and each bounding box is printed.

Each box lists the *x* and *y* coordinates for the bottom-left-hand-corner of the bounding box, as well as the width and the height. The results suggest that two bounding boxes were detected.



We can update the example to plot the photograph and draw each bounding box.

This can be achieved by drawing a rectangle for each box directly over the pixels of the loaded image using the *rectangle()* function that takes two points.



The complete example is listed below.



Running the example, we can see that the photograph was plotted correctly and that each face was correctly detected.

# 

# 

# 

# FACE DETECTION USING WEBCAM

# Using the below snippet we can read live feed from webcam and detect faces

# 

# This is the result of detecting faces from webcam.

# 

# CHAPTER 6

### TESTING

* 1. **TESTING PLAN**

Testing process starts with a test plan. This plan identifies all the testing related activities that must be performed and specifies the schedules, allocates the resources, and specified guidelines for testing. During the testing of the unit each component is tested and the errors are noted. The final output of the testing phase is the test report and the error report.

### Whitebox Testing:

* + 1. **Model Testing:**
       1. We have tested the model using existing malicious sets and arrived at a good accuracy.
       2. Tried using invalid permission sets for the model .

### Api Testing:

* + - 1. Tested the application by trying to post empty data and neglecting the required data fields.
      2. We checked the error identifying capability of the Api by giving false data

### UI Testing:

* + - 1. Checked the functionality of buttons and responsiveness of the user interface

### Blackbox Testing

The application is given to our peers who are not involved in its development and they were asked to try it out and give us feedback of the problems they faced or the errors they encountered while using it. From the given feedback we resolved all the errors.

### Unit Testing:

All the components that made up the application are checked manually to check if they are handling the data being passed on to them.

### Integration Testing:

All the components are Integrated and errors were annihilated by performing tests while integrating them

# CHAPTER 9

### CONCLUSION

As stated in this article, these results are very encouraging as new methods of targeting customers could be derived from this solution. The proposed model consisting of the Improved Markov chain based visualization improves the web analytics by providing accurate visualization of the website clickstream data. This article suggested the method of interactive visualization in order to utilize these results in the analysis of data for different applications.

In the field of clickstream data research is still in its earliest stages, much research still should be finished. With the rebellion of new and speedier innovation, the idea of big data is exceptionally hot right now, particularly on the grounds that companies can, more than ever, make an interpretation of customer data into higher revenue. In the future research, it will be analyzed how to utilize these results for different applications. Likewise the use of new learning algorithms to fit clickstream data, namely, by introducing other models such as neural.

### BIBLIOGRAPHY

**[1].** De Oliveira, M. F., & Levkowitz, H. From visual data exploration to visual data mining: a survey. IEEE Transactions on Visualization andComputer Graphics, 9(3), 378- 394, (2003) Jin Li, Lichao Sun, Qiben Yan, Zhiqiang Li, Witawas Srisa-an, Heng Ye

**[2].** Moe, W. W. An empirical two-stage choice model with varying decision rules applied to internet clickstream data. Journal of MarketingResearch, 43(4), 680-692, (2006).

Zhuo Ma, Haoran Ge, Yang Liu, Meng Zhao, Jianfeng Ma

**[3].** De Bock, K., & Van den Poel, D. Predicting Website audience demographics for web advertising targeting using multi-website clickstream data. Fundamenta Informaticae, 98(1), 49-70,(2010).

**[4]**. Chen, L., & Su, Q. Discovering user's interest atE-commerce sites using clickstream data.In Service systems and service management(ICSSSM), 2013 10th international conference on (pp. 124-129). IEEE, (2013).21. Schellong, D., Kemper, J., & Brettel, M.Clickstream data as a source to uncover con-sumershopping types in a large-scale online setting,(2016). Md. Shohel Rana, Sheikh Shah Mohammad Motiur Rahman,

Andrew H. Sung

**[5]**. Shi, C., Fu, S., Chen, Q., & Qu, H. VisMOOC:Visualizing video clickstream data from massive open online courses. In Visualization Symposium(PacificVis), 2015 IEEE Pacific (pp. 159-166).IEEE, (2015).

**[6].** Brinton, C. G., & Chiang, M. Mooc performance prediction via clickstream data and social learning networks. In Computer Communications(INFOCOM), 2015 IEEE Conference on (pp.2299-2307). IEEE, (2015).

**[7].** Gilks, W. R., Richardson, S., & Spiegelhalter, D.(Eds.), Markov chain Monte Carlo in practice.CRC press, (1995).

**[8].** [http://www.msnbc.com](http://www.msnbc.com/)

**[9].** Steinwart, I., & Christmann, A. Support vector machines. Springer Science & Business Media,(2008)