Automatic Hand Gesture Recognition

Tuesday, 28 April 2015

School of Computer Science,University of Manchester

3rd Year Project Report - 2015

Supervisor - Dr Ke Chen

-by Anant Atray

MEng in Computer Science wIE

Number of words - 7500 words

Abstract

Gesture Recognition is one of the most popular and viable solution for improving Human Computer Interaction and it has become very popular in the recent years due to it s use in gaming devices like Xbox, PS4 and other devices like laptops, smartphones etc . Gesture Recognition and more specifically hand gesture recognition has usage in various application like medicine , accessibility support etc .

This report describes my third year project “Automatic Hand Gesture Recognition “. It discusses the various approaches and techniques that have been used for hand gesture recognition . Furthermore it mentions the various methods used for development and their detailed description , it showcases the results obtained and the testing performed to test the developed software artefact.

As Hand Gesture Recognition is related to two major fields of image processing and machine learning therefore this report also mentions the various tools and APIs that can be use to implement various methods and techniques in these fields.

Acknowledgements

I would like to thank my supervisor Dr Ke Chen for his support and guidance and for making the project interesting and challenging.

I would also like to thank my family and friends for supporting me throughout my studies and project.

List of Figures

List of Tables

Contents

Chapter 1

1. Introduction

A gesture is a spatiotemporal pattern which may be static , dynamic or both,[1] and is a form of non-verbal communication in which bodily motions convey information. Gestures include motion of head, hands, fingers or other body parts.[2] Gesture Recognition collectively refers to the whole process of tracking human gestures , to their representation and conversion to semantically meaningful commands.[3] Gesture Recognition and more specifically hand gesture recognition can be used to enhance Human Computer Interaction (HCI) and improve the effective utilisation of the available information flow.[3]

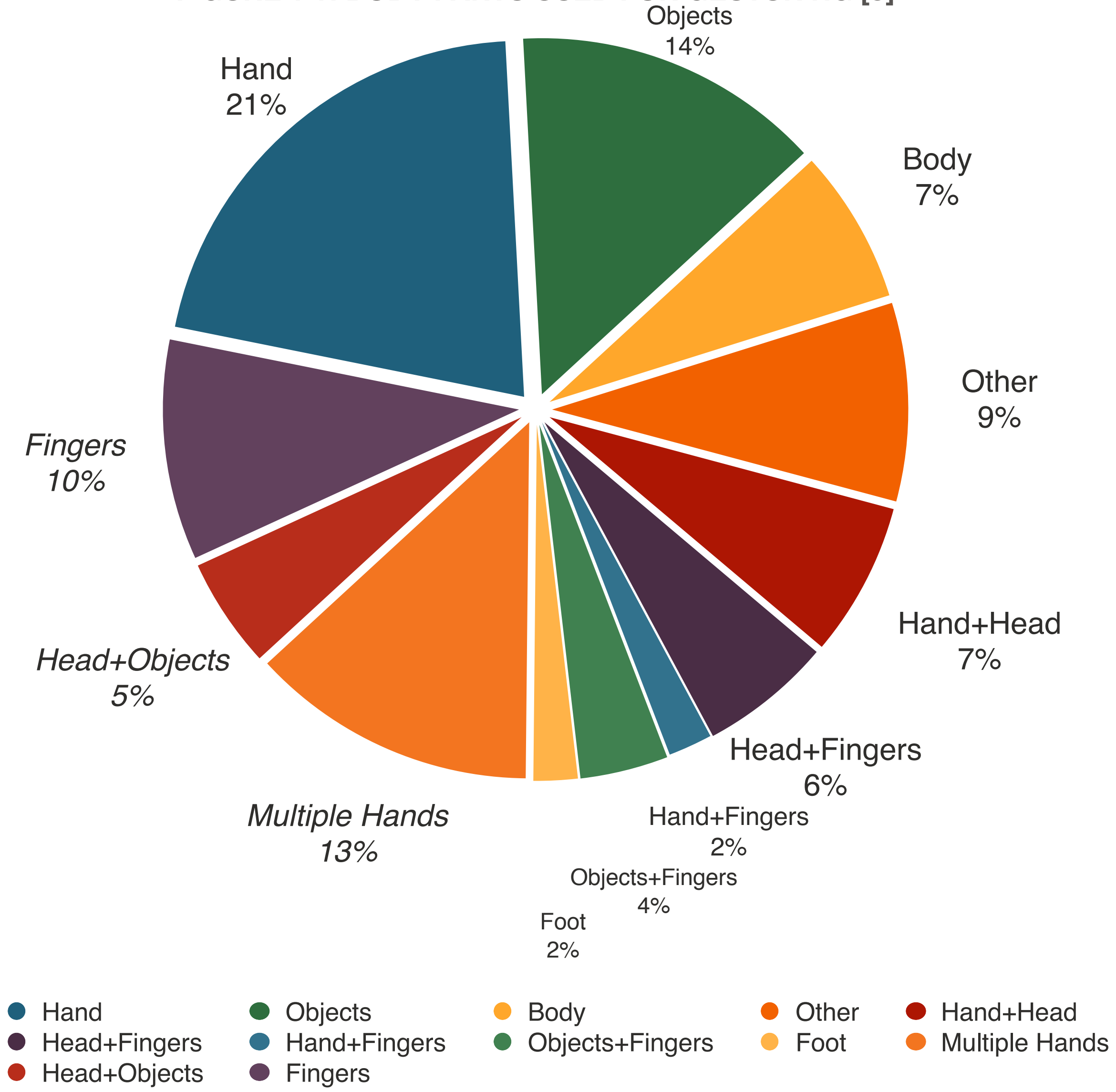
In this chapter we discuss the motivation for this project, the aims and objectives of the project and the overview of this report.

1.1 Motivation

This project is related to two significant fields , computer vision and machine learning . Both of these field are of immense importance in contemporary times due to their widespread use in various disciplines . Computer vision can be defined as a field that incorporates methods for acquiring, processing, understanding and using images and in general any high dimensional real world data in order to produce useful information.[4][5][6] Computer vision has been extensively used in various fields like Human computer Interaction, Medicine, Physics,Image reconstruction etc over the past years and lately it has gained much more traction as it has been used in mainstream devices like Xbox, PS4, smart phones , Tablets , medical devices etc. Machine Learning on the other hand is a subfield of Computer Science that evolved from studying pattern recognition and computational learning in Artificial Intelligence.[7] Machine learning is closely related to computational statistics , prediction making and mathematical optimisation.It has been widely used for applications like spam filtering, Computer Vision, OCR, search predictions and other prediction based applications.[8]

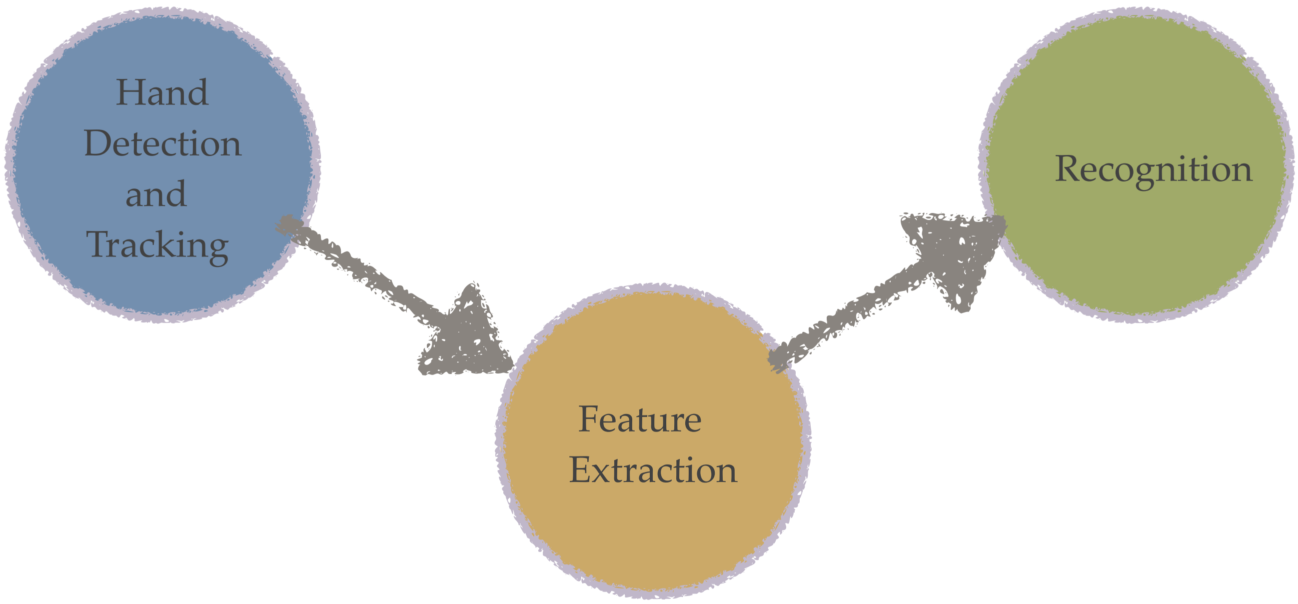
This project focuses on gesture recognition and it uses computer vision and machine learning techniques to achieve this goal.Gesture recognition is important in the field of HCI and HCI plays an important role in applications like Gaming, User Interaction with software systems and accessibility support.We can use various body parts like hand, fingers, head and other objects to perform gestures, but this project focuses on hand gestures because as shown in figure 1.1 hands are used for performing 21% of gestures

FIGURE 1.1: BODY PARTS USED FOR GESTURING [3]



and along with other body parts they are used in a majority of the gesture performed. Historically , different hardware devices like special hand gloves and accelerometer have been used to recognise and track hand gestures but over the past few years camera usage has been the major approach for hand gesture recognition due to natural and non intrusive interaction with the computer system. Various Cameras have been used for this application from normal webcams, stereographic camera, depth sensing cameras , to infra red cameras.In this project we focus on using a normal web camera as they are ubiquitous and therefore provide better chances of acceptance by the general public. Generally, Hand Gesture Recognition involves three major steps as shown in figure 1.2; the first step

FIGURE 1.2: STEPS FOR HAND GESTURE RECOGNITION



is hand detection and tracking which involves capturing the image/video then performing some preprocessing to enable us to detect the hand in the frame and then using other techniques to track the hand in consecutive frames and throughout the video, the next step is feature extraction which involves extracting features from the image that represents important properties of hand gestures and then we use machine learning techniques to use the extracted features to classify the hand gestures.The techniques used in these steps have usage in various other fields like image processing, Data analysis, prediction based applications, image reconstruction etc , so working with these techniques can help in understanding and improving other applications too.

1.2 Aims and Objectives

The aim of the project is to create a software that recognises pre defined hand gestures using various computer vision and machine learning algorithms.As mentioned in figure 1.2 we can divide the project into three major steps which represent the major objectives in the project.

1.2.1 Hand Detection and Tracking

This step deals with detection of hand in the frame and tracking it through the video, our objective in this step is to create a robust system that can detect and track hands of different skin colours in varying light conditions with different but simple background.

1.2.2 Feature Extraction

This step deals with extracting important features that represent important characteristics of the gesture throughout the video and then storing these features. Our objective in this step is to find features that represent shape, motion, size, reflectivity and other important properties. We want features that are generative and not discriminative as this will allow multiple gestures to be recognised with limited features.

1.2.3 Recognition

This step deals with recognising and classifying the performed gesture.It has two phases , the training phase which involves training the system on datasets and the classification phase which involves classifying the performed gestures, our objective in this step is to obtain classification with high accuracy within minimum time.

1.3 Overview of Report

This report is divided into 5 chapters: Chapter 1 is the current one and it gives an introduction about the project, explaining motivations and aims and objectives. Chapter 2 discusses the various methods and techniques that have been used and are currently used for similar hand gesture recognition giving a context to the problem . Chapter 3 discusses the various methods used for development and their implementation . Chapter 4 discusses the software artefact developed and its evaluation through various types of testing.Chapter 5 gives a reflection for the project and a conclusion discussing limitations and future work.

Chapter 2

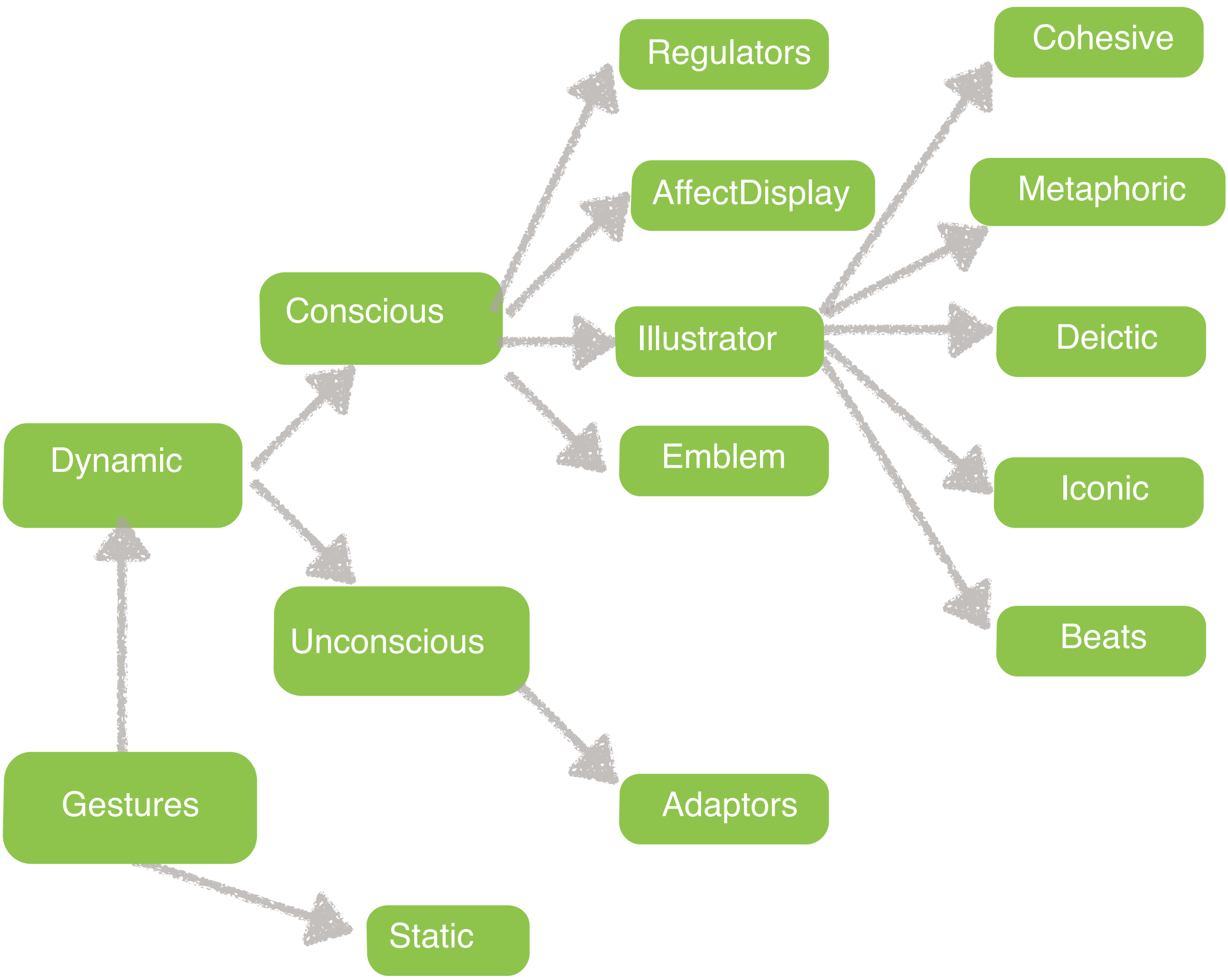
2. Context

This Section discusses in further detail the context and background of the project , giving insights into the field of hand gesture recognition , highlighting popular methods and techniques .

2.1 Hand Gestures

Hand gestures can be broadly divided into two major categories , dynamic hand gestures and static hand gestures . Static Hand Gestures(often referred as hand postures) can be defined as spatial orientation or position of hand in space for a limited period of time[3], like signalling to stop with open palm. Whereas, Dynamic Hand Gestures can be defined as spatiotemporal motion of hands and/or other body parts for a limited period of time , like making a fist or waving a hand . We can further sub-divide these categories into subcategories to get a better classification of hand gestures as seen in figure 2.1.

FIGURE 2.1: VISION BASED HAND GESTURES[9]



Without going into unnecessary details of these gestures ,in this project we will focus on the broad category of conscious dynamic gestures.

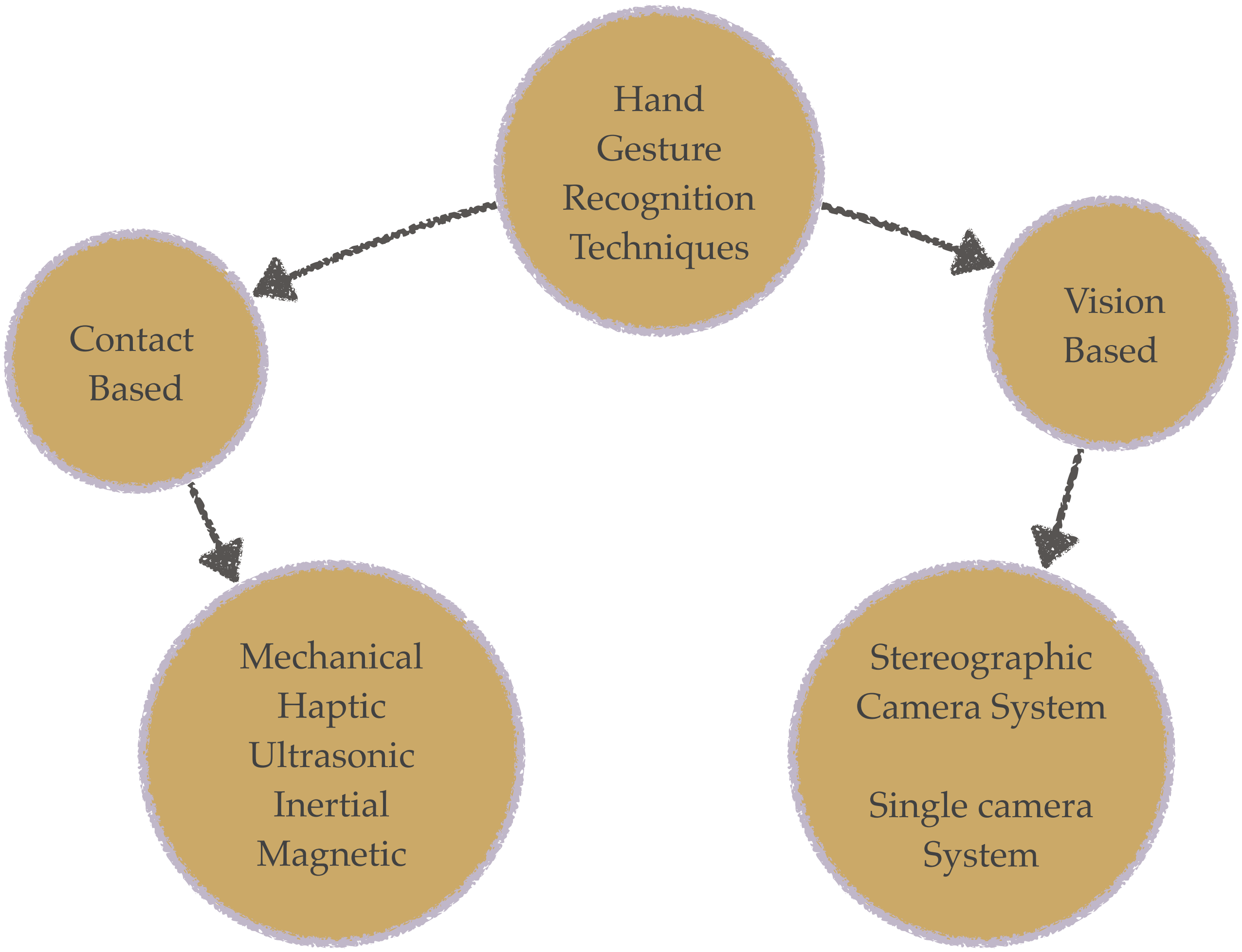
2.2 Hand Gesture Recognition Approaches

There are two major techniques that are used for hand gesture recognition, namely contact based techniques and vision based techniques.[10]

2.2.1 Contact Based

Contact Based techniques are based on physical interaction of the user with a physical device that acts as an input device for the system that recognises gestures.These devices generally have components like gyroscope and accelerometer that measure the forces acting upon the hand and thus inherently try to map the motion of hand. Contact Based devices can be subdivided into five major categories as shown in figure 2.2.

FIGURE 2.2:HAND GESTURE RECOGNITION TECHNIQUES[9]



Mechanical devices include body suits like IGS-190 shown in figure 2.3 and Cyberglove 2 as shown in fig 2.4 , these devices use various sensors for hand gesture recognition . Haptic devices include touch screen devices from phones to tablets[12] to even smartwatches now which also use accelerometer and other technique to improve accuracy.Ultrasonic devices are composed of emitter, reflection disks and sensors, which work together to track gesture using triangulation , speed etc [9] .

FIGURE 2.3:IGS-190 BODY SUIT[11]

FIGURE 2.4:CYBERGLOVE 2 [15]

2.2.2 Vision Based

Vision based techniques are based on interaction of the user with single or multiple camera setups, these cameras can vary vastly in nature from simple webcams, infrared cameras to stereographic cameras. This interaction is used in the form of a video by the system for hand gesture recognition. [13] Vision based devices can be subdivided into two major categories as shown in figure 2.2 . Stereographic camera systems as shown in figure 2.5, use two or more cameras to get depth information along with 2-d information to facilitate 3 dimensional hand gesture recognition.Whereas, single cameras system generally include webcams and infrared cameras that shoot in just two-dimensions.

FIGURE 2.5 : STEREOGRAPHIC CAMERA [16]

2.2.3 Comparison Between Contact and Vision Based

Both of the above mentioned techniques have their advantages and disadvantages as can be seen in table 2.1. Vision Based Techniques are more widely used as the user doesn't need to be trained or be experienced, they don't require any type of user cooperation that might cause any inconvenience to the user. Compared to Contact Based Techniques that may be intrusive like wearing gloves or suits ,Vision Based Techniques are non-intrusive as mostly there is no need for wearing specialised hardware.The major disadvantage for contact based devices are the health hazards they might pose due to intrusive use of specialised hardware like mechanical sensor material which might raise symptoms of allergy, and magnetic devices which might increase the risk of cancer. [17] Whereas, there are no such hazards in vision based techniques. Although Vision based techniques have some disadvantages like occlusion problems, low accuracy in extreme conditions, complex configuration etc , but it is very user friendly and is therefore widely used.

TABLE 2.1 : COMPARISON BETWEEN CONTACT BASED AND VISION BASED DEVICES [3]

	Contact	vs	Vision
User Cooperation	Yes ✗		No ✓
User intrusive	Yes ✗		No ✓
Precise	Yes/No		No/Yes
Configurable	Yes ✓		No ✗
Flexible in Use	No ✗		Yes ✓
Health issues	Yes(No) ✗		No ✓

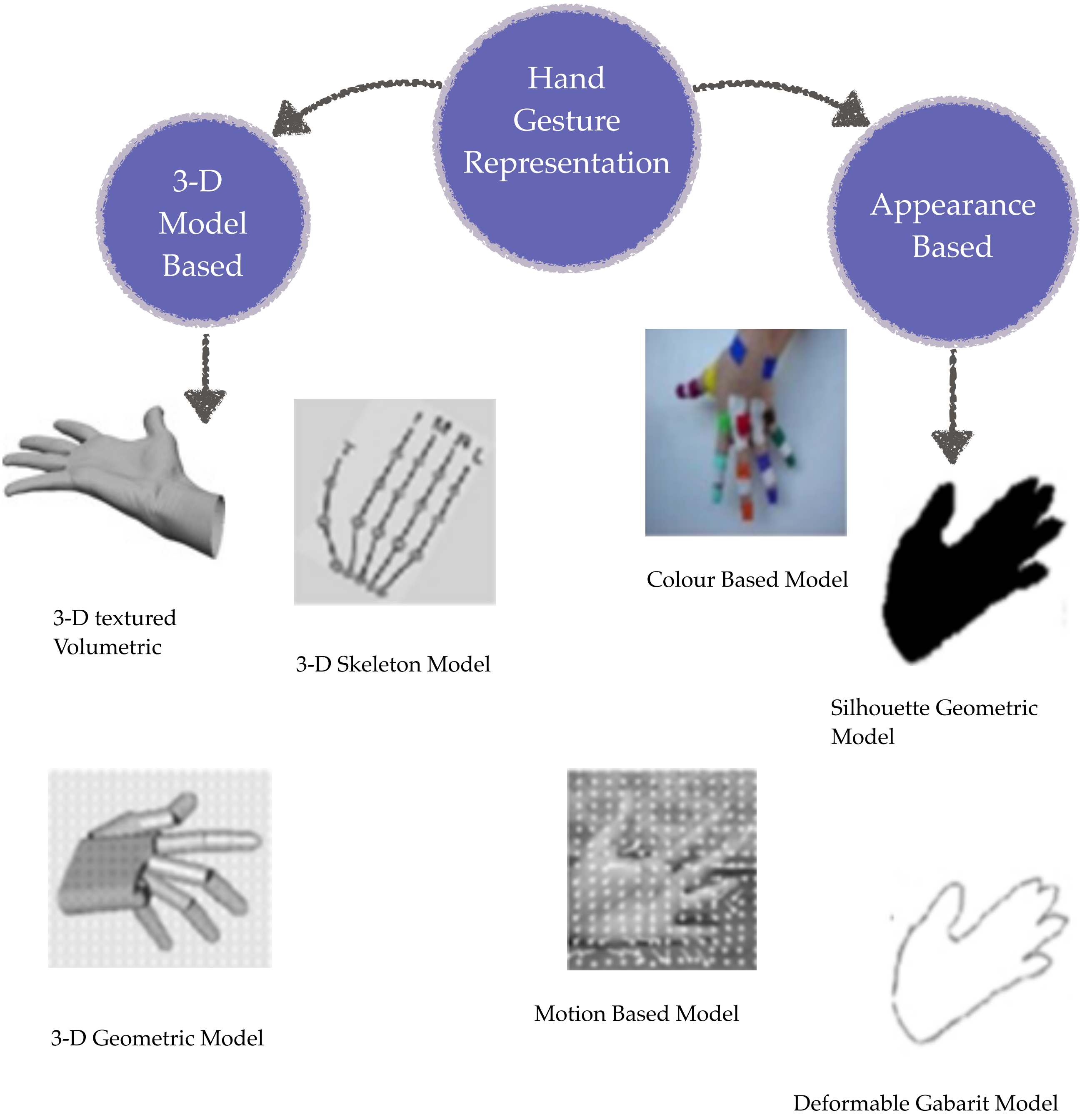
2.3 Hand Gesture Recognition Techniques

This section discusses the methods that have been used widely for the various phases of hand gesture recognition wiz. detection, tracking, feature extraction and recognition ; using vision based techniques.It also discusses the various hand gestures representation and models that have been used in the vision based approach.

2.3.1 Vision Based Hand Gesture Representation

Vision based techniques require abstraction and representation of hand and other body parts to model them, and then using these models to track and recognise these body parts and eventually identify the gestures performed . We can identify two major categories for hand gesture representation namely , 3-D based Models and Appearance Based models.[3] These Models can be further subdivided into various subcategories as shown in figure 2.6. 3-D based models can be represented as geometric model, textured volumetric models and skeleton model, whereas Vision Based models can be represented as colour based models, motion based model, silhouette geometric model and deformable gabarit (french for template or size) model as shown in figure 2.6.[3]

FIGURE 2.6 :VISION BASED HAND GESTURE RECOGNITION [3][24]



3-D based model for hand gesture representation represents the spatial description of the hand in 3 dimensional space. The temporal nature of the hand gestures represented through this model is divided into three phases wiz. pre-stroke , stroke and post-stroke phase.[18] Each of these phase represent changes in the spatial position and temporal state of the hand. Various cameras focus on the hand;then compute parameters and then using these parameters track the hand in space. These cameras update the model parameters along with tracking the hand, resulting in high accuracy but with the cost of being computationally intensive and hence requiring powerful and specialised devices.

Textured volumetric models are highly detailed and contain skeletal and skin information, geometric models are less detailed but have informant skeletal and volumetric information represented through geometric shapes; and skeleton model have the essential skeletal information represented through lines and dots.

Appearance based models on the other hand characterises the hand representation information in two dimensions, it can be divided into two major sub-categories; static based models and motion based models.[3] Static Based models include colour based models which include using hand strips or markers, multi scale colour features and hierarchical models for hand gesture recognition[3][20] ; silhouette based models which include using geometrical properties like shape , centroid, orientation, size, hull , bounding boxes etc[3][21] ; and deformable gabarit models which are based on contours which represent the shape and template of the hand. [3][21]On the other hand , Motion based models use the motion of the hand in consecutive frames to detect the hand in image sequences.[3][22]

In this project we will be using appearance based approaches as they don't require specialised devices and are computationally non-intensive .

2.3.2 Detection Phase

This section discuses the various techniques that have been widely used for the detection phase of hand gesture recognition, this phase deals with detection of the hand in the frame and creating a segmented image to represent the detected hand.It is pivotal for this phase to be robust to different lighting conditions , skin colours and background as it deals with detection of hands, which is the basis for the other phases that follow this phase.Some of the widely used methods use features like motion, colour and shape .

2.3.2.1 Colour

Skin Colour Segmentation is a widely used techniques for detecting hands in the frame. Detection of the skin colour is highly influenced by the colour space used for the image , several colour spaces have been used like RGB, LAB, HSV , YCrCb etc .[3][24] Colour Spaces like HSV, YCrCb and Lab,[3][24][25][26] that separate the chromaticity from luminance and use chromaticity rather than colour of skin are desirable as they are robust to lightning conditions and shadows.[3][24][25][26][27]Since skin colour varies from person to person and lighting /shadow condition and camera properties introduce variability in the frame , therefore we need to use some methods for tackling these problems.[3][24][25][26[29][30]Methods like invariant representation of skin colour with respect to changes in illumination, and estimating new parameter for mean and covariance of multivariate gaussian colour distribution etc [3][24][25][26][27][28] have been tried but they are still affected by changing light conditions. There are other challenges like having a background of the same skin colour too. We can use background subtraction for such cases.

2.3.2.2 Motion

Motion detection of an object can be used to track the moving object in consecutive frame, but we need to make few assumptions to employ this technique, we assume that the only object moving in the frame is our hand , which is seldom the case,also with motion of our hands the shadows also move .[31][3] To tackle such problems we can use motion detection along with other information like background value etc, we can use the fact that the pixel values for the background will not change vastly as compared to the pixel detecting the moving hand.[3][32]

2.3.2.3 Shape

Shape of the hand can be used to detect hands in various ways, we can obtain information by calculating contour which is similar to the shape of the hand and this makes it independent to lighting, skin colour, shadows etc [3] but defining shape of the hand is a very challenging problem because it varies between gestures and different point of views. Moreover techniques used for calculating contours like edge detection [3][33] not only calculate edge of the hand but also other objects in the frame making it necessary to use other techniques to make sense of this edge information like local topical descriptors and even multiple cameras in some cases.[3][34] A

2.3.3 Tracking Phase

After detection of the hand the next step is to track this hand in consecutive frames of a video.This section discusses the techniques that have been widely used for tracking the hand throughout the video sequence. Using a tracking techniques with real time performance is important to track the hand in real time with high accuracy.

2.3.3.1 Template Based

Template based methods have some similarity to the shape based detection methods used for hand detection. We can divide these methods into two broad categories Correlation based tracking and contour based tracking.[3][35]

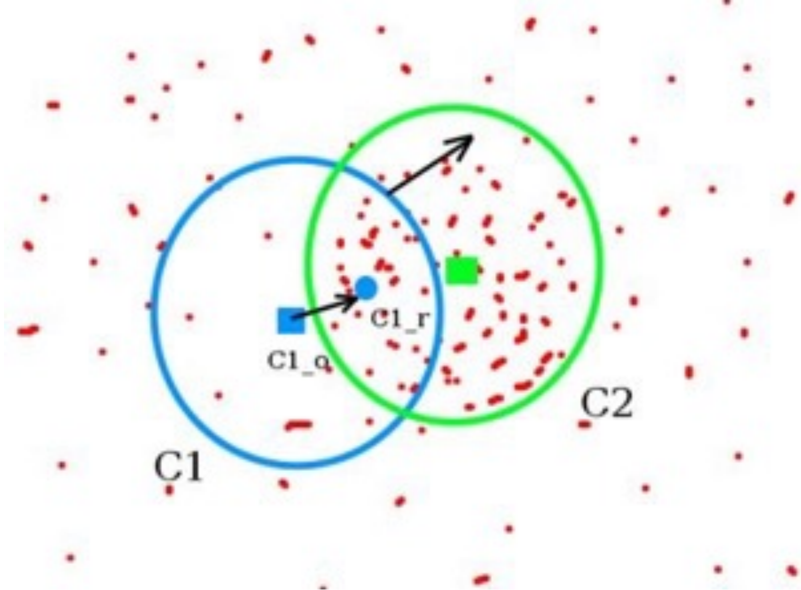
In correlation based tracking template of the hands are used for tracking the hand, we use the region where we have detected the hand to make templates and then using this template in the next frame we look in the vicinity of the previous position of the hand to find the hand.[3][36]Few approaches detect hand as blobs in the image sequence and correlate blob in vicinity to each other to make a trajectory and thus track the hand.[3][37][39]

In contour based tracking deformable contours are used for tracking hands in consecutive frames.[3][38] We calculate the contour after detection and preferably on a segmented image which gives us a boundary of the hand along with other regions, we can then use properties like, size, area, centroid etc to differentiate between various contours to pick the one that represents the hand.

2.3.3.2 Meanshift and Camshift

Meanshift is a non-parametric feature space algorithm for calculating maxima of a density function.[40]It uses density based models to track targets which in this case is the hand, we define the size and location of the tracking window in the frame , meanshift iteratively tries to overlap the centre of this window with the centre of the area of maximum pixel density as shown in figure 2.7.[44] Mean shift has been widely used due to low computational cost and speed,[3][41] but it is limited in functionality as the size of the tracking window is not variable which makes it inaccurate in situations with scaling.[3][42]

FIGURE 2.7 :MEANSHIFT ALGORITHM VISUAL[44]



Camshift (Continuous adapted Meanshift) which is based on Meanshift on the other hand changes the size and distribution pattern while tracking therefore overcoming the limitation of Meanshift. Meanshift and Camshift perform well in situation with simple background but are inaccurate in situations with complex background. These tracking are often used with other techniques to create robust systems.[3][43]

2.3.4 Feature Extraction

Detection and tracking phases allow us to detect and track the hand in the video sequences, but in order to recognise and classify the hand gesture we need to extract features that represent useful information like shape, size, motion, form etc . Good features should be independent of scaling, rotation and translation , they should be easily computable and should represent unique information. This section discusses few of the features that have been generally used for hand gesture recognition.

2.3.4.1 HuInvariant Moments

An Image moment can be defined as a particular weighted average (moment) of the image pixel intensity or a function of such moment. [46] Hu Invariant moments came up in 1960s when Hu derived them using algebraic invariants ,[47] since then they have been extensively for shape description and recognition as they are independent to scaling, rotation and translation. [48]2-dimensional moments of an M × M image that has grey function f (x, y), (x, y = 0, … M − 1) is given as,

$$m_{pq} = \sum_{x=0}^{x=M-1} \sum_{y=0}^{y=M-1} (x)^p (y)^p f(x,y)$$

When a scaling normalisation is applied the central moments change as,

$$\mu_{pq} = \sum_{x} \sum_{y} (x+a)^p \cdot (y+b)^q f(x,y)$$

$$\eta_{pq} = \mu_{pq} / \mu^{\gamma}_{00}, \quad \gamma = [(p+q)/2]+1$$

Hu invariant are seven values computed by normalising central moments through third order.They are given as following in terms of central moments

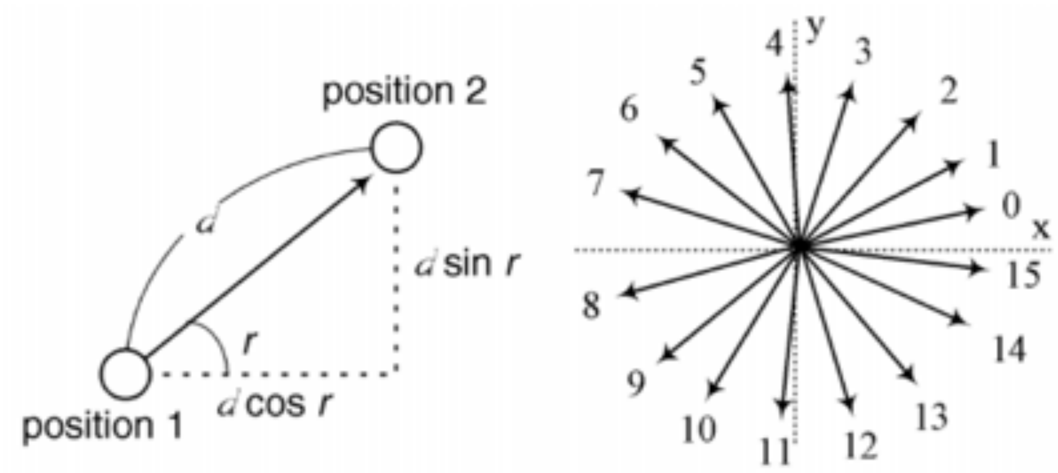
$$M1 = (\eta_{20} + \eta_{02}),$$
$$M2 = (\eta_{20} - \eta_{02})^2 + 4\eta^2{}_{11},$$
$$M3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2,$$
$$M4 = (\eta_{30} + \eta_{12})^2 + (\eta^{21} + \eta_{03})^2,$$
$$M5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2],$$
$$M6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}),$$
$$M7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} + 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2].$$

Each of these moments represent something or another like M1 is analogous to moment of inertia around's image centroid , and M7 is skew invariant which differentiates between mirror images, similar each of these have such function , but they are not perfect, for example M3 is dependent on other moments etc .[48] So , basically if we take an image and then rotate it ,scale it and translate it , we should still get the same values for the hu invariant moments.

2.3.4.2 Motion Analysis

Both of the methods discussed above focus on extracting features using shape and such structural information , motion analysis on the other hand focuses on temporal information for feature extraction . We can have various techniques for motion analysis but one of the widely used one is using orientation of the hand in consecutive frames to get an idea of the trajectory of motion. In such a techniques we can use centre of the hand in each frame and use its position for making some sense of the trajectory, but a better way is use to calculate position of the centre in each frame and then calculate the tan of the angle between the horizontal line and the line joining the centres in consecutive frames to get an idea of direction of the motion of the hand as shown in figure 2.8

FIGURE 2.8 :HAND ORIENTATION THROUGH ANGLE CALCULATION[49]



2.3.5 Recognition

After feature extraction the next step is to use these features to recognise and classify the hand gestures, this section discusses some widely used techniques for recognition of hand gestures. Since in this project we are working on recognising conscious dynamic gestures , therefore we focus on techniques which can handle the temporal nature of such dynamic gestures

2.3.5.1 Hidden Markov Models

Hidden Markov Models (Hmms) are one of the most widely used methods for recognition in applications like speech recognition , handwriting recognition , gesture recognition etc .[50] Hmm can be considered as a generalisation of markov chain without its restriction of having only one transition arc.[3][53] We can define the chain rule as shown below which is used to calculate joint distribution of any variable from set of random variables.

$$P(\cap_{k=1}^{n}, A_k) = \sqcap_{k=1}^{n} P(A_k | \cap_{j=1}^{k-1}, A_j)$$

Markov Property essentially states that the future is independent of past , given the present i.e probability of a variable depends only on the immediately preceding variable and not other preceding variable , mathematically shown below

$$P(\cap_{k=1}^{n}, A_k) = \sqcap_{k=1}^{n} P(A_k | A_{k-1})$$

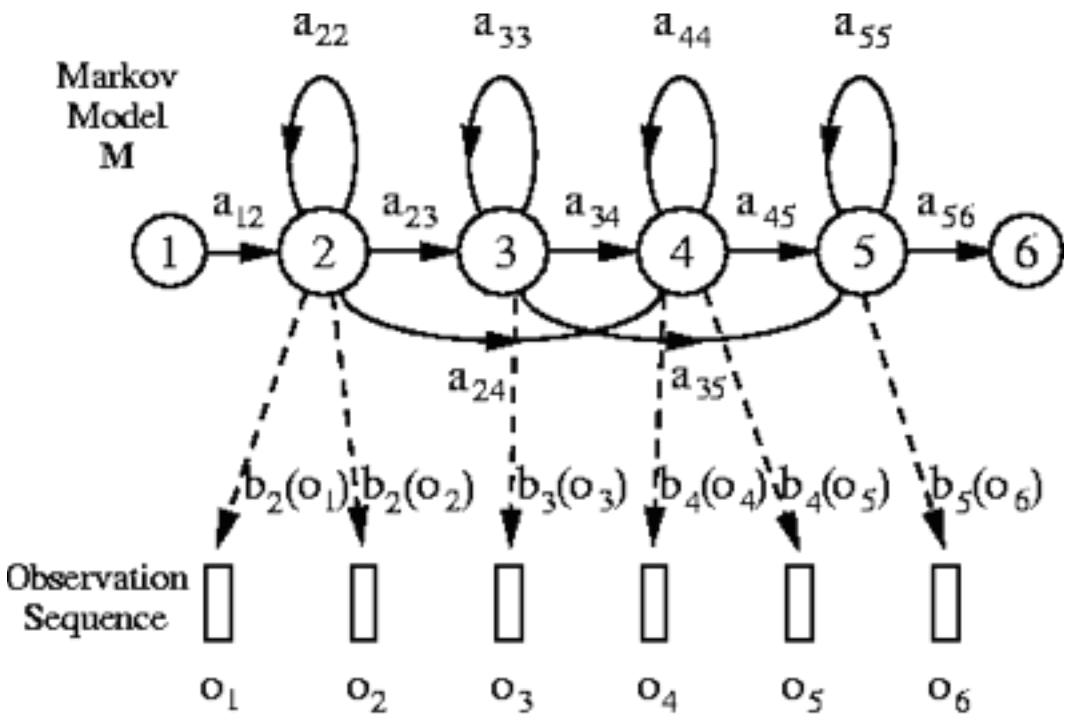
This represents the idea behind markov model , which are essentially simple finite state automata in which each state transition has a related probability value and a single transition arc making them deterministic. Since , Hmms can have multiple transition arcs therefore they are non-deterministic and it is not possible to determine the state sequences for an input simply by looking at the output.[3]In Hmms the state are not visible/ observable hence the word hidden, only the output that is dependent on the state is visible.

Generally Hmms are defined as a quintuple $\lambda = (N,M,A,B,\pi)$.[50][51][52] N is the number of states in the model, M is the alphabet size i.e the number of observation symbols per state, A is a NXN transition matrix giving a probability distribution of transition between states, B is a NXM emission Matrix giving a probability distribution of observation of symbol from each state and π is the prior probability matrix(vector) giving initial probability of each state. In hand gesture recognition , each state of hmm can represent a hand posture, state transitions probabilities can represent the probability that one hand posture leads to some other hand posture and an output symbol represents a posture and a sequence of output symbols represent a hand gesture.[3]Since hand gestures eliminate the possibility of sudden changes in posture of hand due to laws of physics , therefore jump from state to state in Hmms should be avoided , thus we use forward Hmms which only allow transition from one state to the forward states.[3] [51]

There are three main problem related to Hmms. First is evaluation , in which we have a hmm and an output sequence and we calculate the probability of the sequence being produced by the model, second is decoding in which we have a model and observation sequences and we calculate the most likely sequence of state that produces it, and the third is learning in which we have set of observation sequences and we calculate the model parameter to maximise the probability of the sequence originating from the model.[51][52]

Various Algorithms like Viterbi Algorithm and forward algorithm are used for solving evaluation.Viterbi can also be used to solve decoding and a popular algorithm for solving learning is baum-welch algorithm.[51]

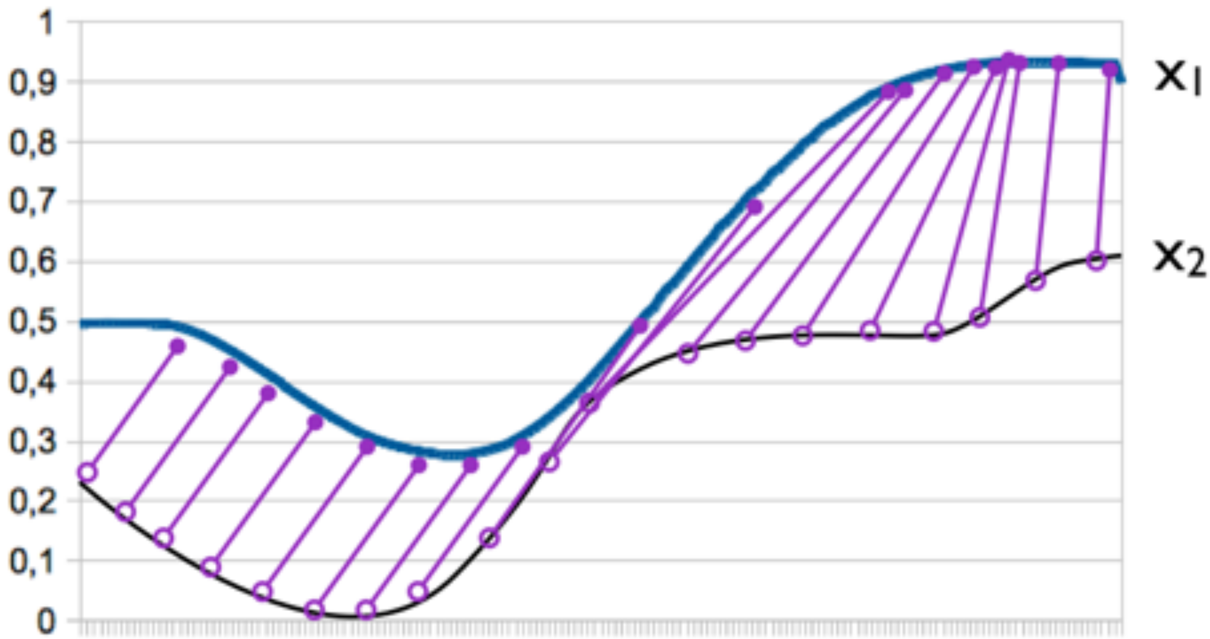
FIGURE 2.9:HIDDEN MARKOV MODEL[53]



2.3.5.2 Dynamic Time Warping

Dynamic Time Warping has been widely used in field like speech recognition , data mining , and motion recognition .[55] It is suitable for data that varies in length with time , it is suitable for hand gesture recognition as it is temporal and thus the duration of the same gesture varies from person to person.The dynamic time warp algorithm calculates distance between two points lying on each signals in terms of their associated value.[3][56] It uses this distance to create a cumulative distance matrix and the path with the least value which represents the ideal warp, which can be understood as syncing between the two signals with minimised distance between the synchronised points .[56] Dynamic time warping works with continuous data and therefore high number of possible hand gestures makes it computationally extensive and unsuitable for real time performance. In simple terms , it tries to warp a signal to match similar points in each signals while following some rules to obtain a signal that matches or tries to match the given signal with minimum distortion.Figure 2.10 gives a graphical representation of dynamic time warping.

FIGURE 2.10:DYNAMIC TIME WARPING[57]



Chapter 3

3. Development

This section discusses the various choices made during the development of the project and the implementation of the various methods used during the project.

3.1 Design

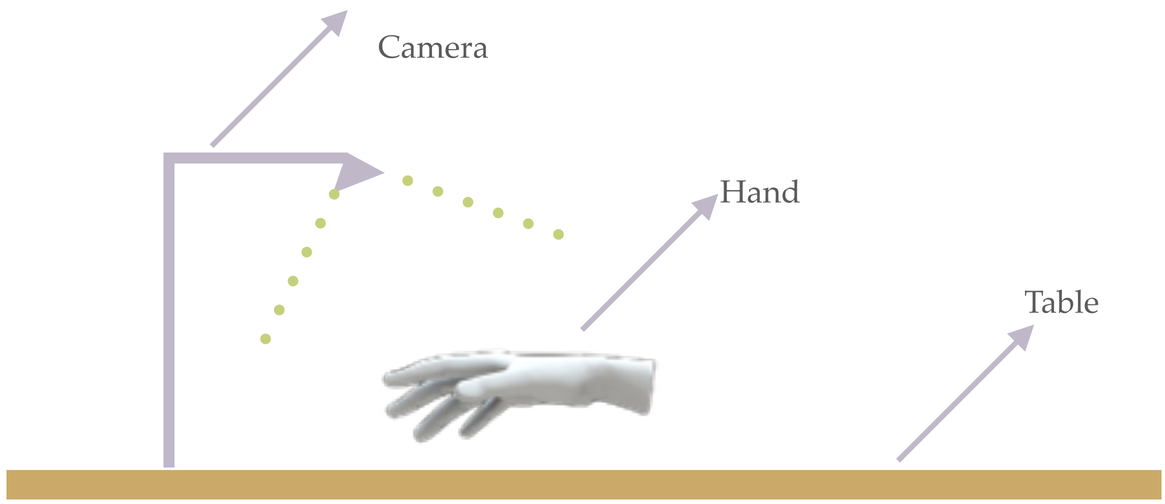
In order to develop a software artefact for the project certain design choices regarding the nature of the project had to be made; these include the assumptions made, APIs used, datasets used,method used for various phases and testing techniques.This section discusses these aspects of the project.

3.1.1 Assumptions

Hand gesture recognition is a complex problem due to large variety of gestures, spatial and temporal variation, multiple degree of freedom, complex and cluttered background, extreme lighting conditions and because of the need to obtain a robust, real time performing system.In order to make the project achievable few assumptions were made.

We decided to use a simple background, thus making it easier to detect and track the hand; to perform gestures in 2-dimensional space to reduce the degree of freedom and therefore complexity; to use a single camera setup that focuses on the hand lying on the table as shown in figure 3.1 thus making the hand a major part of the frame; and we decided to recognise limited hand gestures.

FIGURE3.1:PROJECT SETUP



3.1.2 Dataset

After making the assumptions above the next step was to choose a dataset that satisfies these assumptions and also provide a large number of samples with variation in lighting conditions, skin colour and motion of the hand.As seen in Table 3.1, various datasets were considered , we were looking for a dataset that suited our requirement and had been widely used so we chose the Cambridge Hand gesture Dataset ,as seen at bottom of table 3.1 it has been widely used and as shown in figure 3.2 it suits our assumptions.It has 9 gesture class , with each class having 100 image sequences (5 different illuminations x 10 arbitrary motions x 2 subjects).In this project we are using four classes for hand gesture recognition which are shown in figure 3.3. we are using the first , fourth , sixth and the ninth gesture as shown in figure 3.2

TABLE 3.1 : DESCRIPTION OF DATASETS FOR HAND GESTURE RECOGNITION [58]

Name	Year	Citations (03.02.2014)	Sensors placement	Information	Types of gestures	Purposes & Description	Availability
3DIG [25]	'13	1	Environment	2	Iconic	**Recognition** of iconic gestures where subjects were free to perform their own gesture to depict each object	P
ASL Dataset [21]	'13	-	Environment	5	Sign language	American sign **recognition**. Evaluation of hands **detection** & **tracking.** Acquisition still on-going.	NY
CGD2013 [22] **ChaLearn Dataset**	'13	2	Environment	5	Metaphoric	Multimodal gesture **recognition** of cultural Italian gestures accompanying speech. **Challenge**-related dataset	P
ChAirGest [6]	'13	1	Env. & wear.	5	Iconic & metaphoric	Gesture **spotting** & **recognition** from multimodal data in the context of close HCI. **Challenge**-related dataset	PR
SKIG [26]	'13	5	Environment	3	Iconic & metaphoric	Improve gesture recognition from RGB-D data, notably with different illuminations. Hand gesture **recognition** seen from above	P
6DMG [27]	'12	2	Env. & wear.	5	Iconic & metaphoric	Explore gesture **recognition** from implicit & explicit data. Subjects performed the gestures with a Wiimote in their right hand	P
MSRC-12 [19]	'12	21	Environment	5	Iconic & metaphoric	Gesture **recognition** from the skeleton data. Study the motion variation across users with skeleton data	P
G3D [24]	'12	7	Environment	5	Iconic	Gaming actions and gestures **recognition** & **spotting.** Specifically designed to improve gaming without controller	PR
MSRGesture3D [28]	'12	17	Environment	3	Sign language	Sign language **recognition** from hand depth data. Only the segmented hand sections of the images are provided	P
CGD2011 [29] **ChaLearn Dataset**	'11	17	Environment	5	Iconic & metaphoric	Improve one-shot learning for **recognition**. **Challenge**-related dataset. The competition had a large success.	P
NATOPS Aircraft Handling Signals Database [30]	'11	26	Environment	5	Metaphoric & symbolic (Real vocabulary)	Body-and-hand **tracking** & gesture **recognition** requiring both body and hand information to distinguish gestures	PR
NTU Dataset [31]	'11	68	Environment	2	Metaphoric & symbolic poses	Hand pose & shape **recognition** in cluttered conditions. Only contains static images, no motion.	P
Keck Gesture Dataset [23]	'09	153	Environment	4	Metaphoric & symbolic (Real vocabulary)	Military gestures performed with perturbations in the background. Designed to evaluate gesture **recognition** and **spotting** in harsh conditions.	P
ASLLVD [32]	'08	17	Environment	4	Sign language	A reference database in automatic sign language **recognition** and **spotting** with data captured from several viewpoints.	PR
CHGD [33] **(Cambridge Hand Gesture Dataset)**	'07	136	Environment	4	Metaphoric	Hand segmentation & gesture **recognition** in varying illuminations conditions. It only contains sequences of images.	P

FIGURE3.2:CAMBRIDGE HAND GESTURE RECOGNITION DATASET[59]

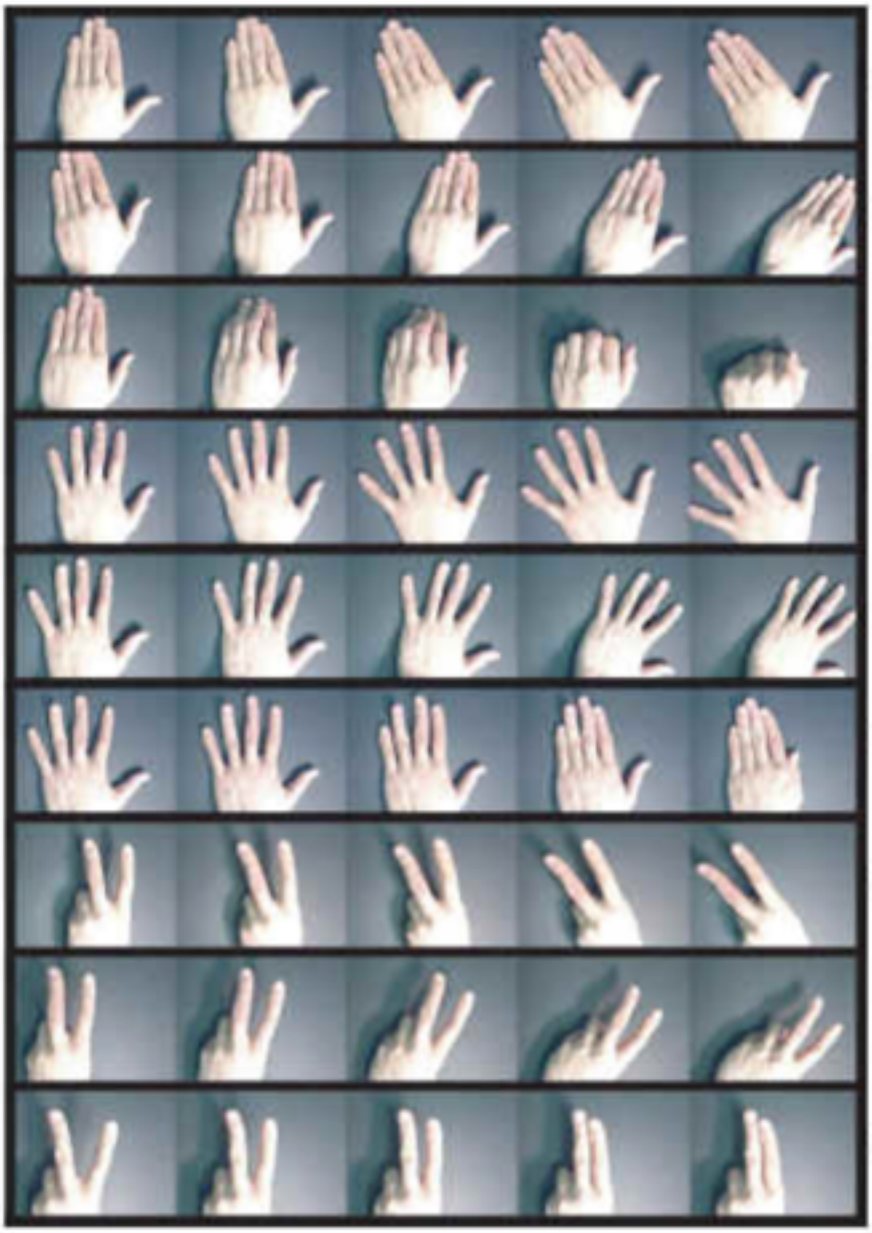


FIGURE3.3:SELECTED HAND GESTURES FOR RECOGNITION

3.1.3 Methods Used

After the initial assumptions and finding the dataset that fulfils them , the next step was to decide the various methods to be used for each phase of the project namely detection and tracking, feature extraction and recognition. This section discusses the methods that were used for each of these phases.

3.1.3.1 Detection

Out of the various methods discussed for detection of hand in section 2.3.2 , methods that were suitable for the project were detection through skin colour and through motion. As seen in figure 3.2 , the dataset has a simple background with a prominent hand in the frame which makes it suitable for skin colour detection , since the lighting conditions vary therefore instead of using the RGB colour space we used the Lab colour space[64] which separates luminous from colours making it more robust to varying lighting conditions , in Lab L stands for lightness. a represents colour between red and green and b represents colour between yellow and blue, we will discuss more about Lab in implementation section.

Apart from this, since the only subject moving in the frame is the hand therefore we also decided to use motion for detection . we will discuss it and its result in implementation section.

3.1.3.2 Tracking

After the detection phase we should obtain a segmented image showing the detected hand, in order to track this hand we decided to use two of the various techniques mentioned in section 2.3.3; Meanshift algorithm and contour based tracking.

Meanshift algorithm is suitable as the motions performed are in 2-d , thus minimising the factor of scalability and since after detection we have a segmented image showing hand region with white pixels therefore Meanshift was suitable for tracking this pixel dense area representing our hand.

The other techniques used is contour based tracking , since the only region with skin colour in the frame is the hand and since the image is segmented showing hand with white region, therefore finding contours and using properties like area is useful in tracking the hand in consecutive frames ,we will discuss this in detail in the implementation section.

3.1.3.3 Feature Extraction

After detection and tracking the next phase is to extract important features to use them for recognition and classification.For feature extraction we decided to use two of the various techniques mentioned in section 2.3.4 ; namely the hu invariant moments and the orientation of the hand.

As seen in figure 3.2 , our dataset has rotation, translation as well as some scaling in the hand gestures , therefore hu invariants are suitable for such a situation , moreover in order to capture the motion trajectory of the hand , we use the orientation of the centre of the hand, by calculating tan of the angle between horizontal line and line connecting centre of the hand in consecutive frames and using this value as a feature. We calculate these features and store them in a matrix.we will discuss more about this in the implementation section

3.1.3.4 Recognition

After detection and tracking, and feature extraction , the next step is to use these extracted features to recognise and classify these gestures.We decided to use the Hidden Markov Models out of the various methods discussed in section 2.3.5 due to the following reasons.

Hand gesture recognition is temporal in nature and Hmms have been widely used for classifying temporal data like speech , handwriting and gestures.[52]Our dataset has unlabelled data and therefore it will be suitable to use unsupervised learning, Hmms provide unsupervised learning making it easier to work on unlabelled data and gives us the flexibility to expand our software to recognise multiple classes with minimum effort. Apart from this Hmms have been widely used and therefore there is a lot of information available about them through various sources.

Now this phase has two parts the learning phase and the classification phase, we will discuss them in detail in implementation section, but here is a brief overview; we use the hand gesture image sequences (actually the feature extracted from them) from dataset to train each Hmm on a particular class of gesture , we have one Hmm for each class we want to recognise, this is the learning phase.The Classification phase uses the trained Hmms to classify the incoming vector of features as one of the class of the various classes that the different Hmms were trained on.

3.1.4 Devices, APIs and Language

After deciding the method to be used for each phase , the next step is to decide the APIs and the languages to be used. This section discusses the devices , APIs and the languages used.

As explained in section 3.1.1 and shown in figure 3.1, we used a simple laptop webcam as our choice of camera , the software was written and tested on Mac OS.

As the project requires substantial amount of image processing and machine learning therefore few of the possible choices for the language to be used were Matlab, Python, C++ ; out of these I decide to use python as I will explain in the next paragraph.

In order to implement various methods for detection and tracking and feature extraction we need to use some API that provides image processing functionality and one of the most widely used API for it is OpenCV[60], it is an API that is written in C with bindings for Java , Matlab and python.Therefore we decided to use OpenCV. Also we can argue that using OpenCV binding with Matlab is much more difficult and its not suitable for real time results , therefore to make coding easier, as there is a lot of resource available for python binding of OpenCV and to achieve real time performance we used python [61].

Apart from all the image processing , we also need to implement Hmm to enable us to recognise the gestures, there are few APIs available for python to implement Hmm like GHMM, hmmlearn, scikit learn. We decided to use the sklearn.hmm module version 0.16.1 [62]as it provides the functionality we want for our Hmm and is much more comprehensive than the others mentioned.

3.1.5 Software Design

Since we have used python in our project , therefore it was much more sensible to use scripts that perform certain tasks as compared to creating object, Thus we made 5 major classes which have python functions that perform certain tasks through the different phases of the project. Each class has specific function . for example detection class has functions for skin segmentation and thresholding to detect the hand, tracking class has function for tracking the detected hand in frame and extracting features from the tracked hand and storing them in vectors,Classify class has function for classifying the gestures based on their score with the trained Hmms , video class has function to use function from detection and tracking and track the hand in real time and extract features and finally we have GUI class that has function to use the system and to display real time tracking and recognition results.

3.2 Implementation

After deciding the methods to be used for each phase and APIs and language to be used for implementing them the next step is to implement these methods. This section discusses the implementation details of the various method used for various phases.

3.2.1 Detection

This section discusses the implementation details of the methods used for detection.As mentioned in section 3.1.3.1 we decided to use skin segmentation and motion to detect hand.

Lets discuss motion, To use motion to detect the hand,we assume that $F_i(x,y)$ is the with frame of the image sequence and $D_i(x,y)$ is difference between i^{th} and $(i + 1)^{th}$ frame

$$D_i(x, y) = T_i\{|F_i(x, y) - F_{i+1}(x, y)|\}$$

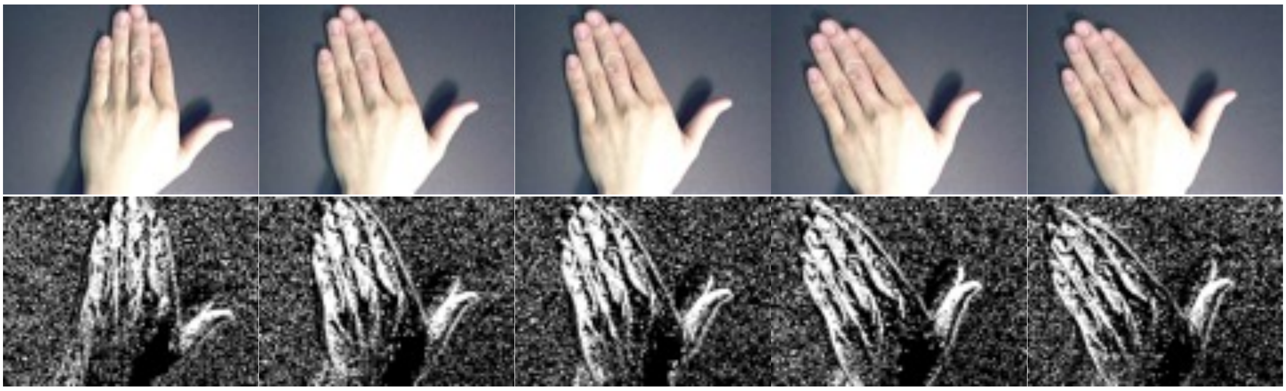
where Ti is the thresholding function , we calculate the threshold value using average luminous of $F_i(x,y)$; i.e $t_m = 0.2\mu$, where μ is average luminous of the image.[45]We calculate average luminous using the equation seen below.

```
averageLuminosity = 0.299*Red + 0.587*Green + 0.114*Blue
threshold = 0.2*averageLuminosity
imageDifference = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
                  - cv2.cvtColor(nextimage,cv2.COLOR_BGR2GRAY)
ret,thresh = cv2.threshold(imageDifference,threshold,255,cv2.THRESH_BINARY)
```

Using this technique we get the following result , as seen in figure 3.4 and figure 3.5,, the detection is not very accurate, we tried various values for threshold but the results were not accurate , this may be due to the interplay of shadow which substantially changes the background , also the whole hand is not in motion making it difficult to detect it, therefore we decided not to use this method in the software .

FIGURE3.4:MOTION BASED DETECTION OF DIFFERENT GESTURES IN DIFFERENT LIGHT

FIGURE3.5:MOTION BASED DETECTION OF A GESTURE



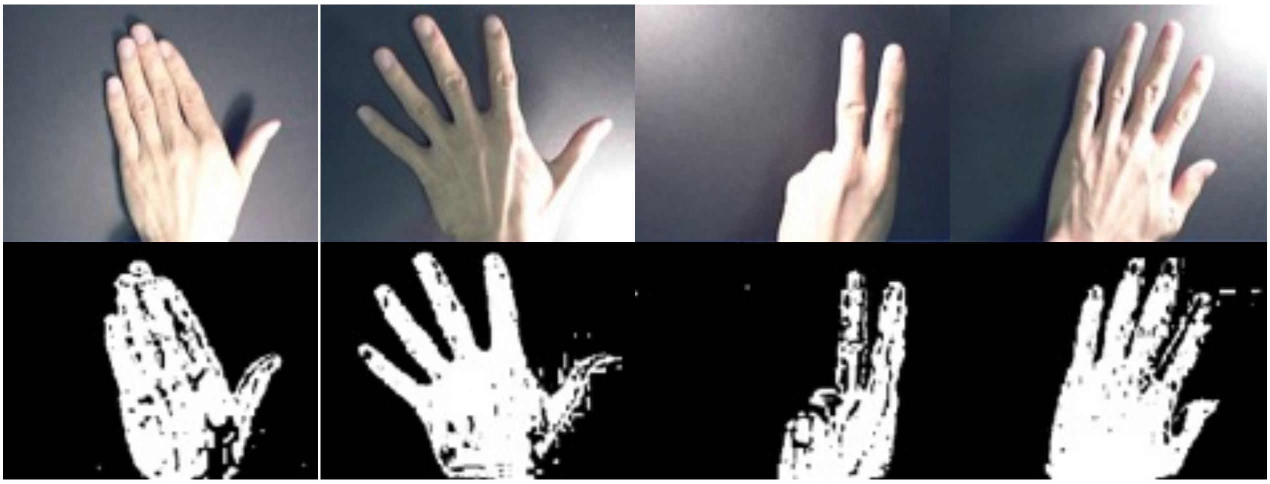
The other method we used was skin segmentation, as mentioned in section 3.1.3.1 we used the Lab Colour space to instead of RGB colour space for skin detection this is because Lab divide the image into three channels L,a and b as mentioned in section 3.1.3.1 we can use a and b channels to recognise colour with out them being affected by lighting condition [63] [64] We start by changing the colour space of the image to Lab using OpenCV[65] and then use the values of channels a and b which are 8 bit channel and thus have value from 0 to 255, to create threshold values to use for thresholding the image and produce segmented image showing the hands as shown below.

```
image = cv2.cvtColor(image,cv2.COLOR_BGR2LAB)
blur1 = image[:,:,1]
blur2 = image[:,:,2]
ret1,thresh1 = cv2.threshold(blur1,124,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)
ret2,thresh2 = cv2.threshold(blur2,128,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)
thresh = thresh1 &thresh2
```

We calculate these threshold values based on the widely used values for depicting skin colour in Lab colour space and on experimentation. [63] [64] Figure 3.6 and 3.7 shows results obtained from this method.

FIGURE3.6:SKIN SEGMENTATION HAND DETECTION OF A GESTURE IN SAME LIGHT

FIGURE3.7:SKIN SEGMENTATION HAND DETECTION OF GESTURE IN DIFFERENT LIGHT



Since we obtain good results therefore we use the skin colour segmentation method in the software.

3.2.2 Tracking

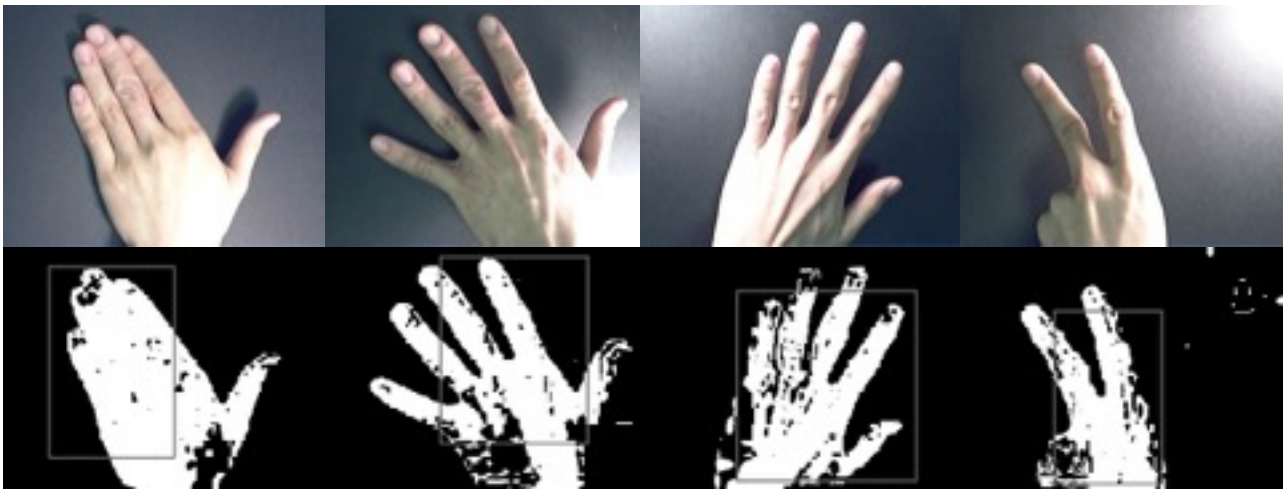
After detection the next step is tracking, in this section we discuss the implementation details of the tracking methods. As mentioned in section 3.1.3.2 we decided to use Meanshift algorithm and Contour Based tracking.

Lets discuss Meanshift, as mentioned in section 2.3.3.2 we use the segmented image obtained from detection phase , then create a tracking window and apply Meanshift Algorithm using OpenCv [66]as shown below

```
# setup initial location of window
track_window = (c,r,w,h)
print '########################################'
# set up the ROI for tracking
roi = frame[r:r+h, c:c+w]
mask = cv2.inRange(roi, 0, 255)
roi_hist = cv2.calcHist([roi],[0],mask,[255],[0,255])
cv2.normalize(roi_hist,roi_hist,0,255,cv2.NORM_MINMAX)
# Setup the termination criteria, either 10 iteration or move by atleast 1 pt
term_crit = ( cv2.TERM_CRITERIA_EPS | cv2.TERM_CRITERIA_COUNT,100000000, 10000000)
dst = cv2.calcBackProject([frame],[0],roi_hist,[0,255],1)
# apply meanshift to get the new location
ret, track_window = cv2.meanShift(dst, track_window, term_crit)
print '########################################'
print index
print '########################################'
print track_window
c,r,w,h = track_window
```

Using Meanshift algorithm we get the result as shown in figure 3.8 where grey window is the tracking window, the tracking is not very accurate and is often not able to track the whole hand, the reason may be the large size of hand in image frame that makes it difficult for the tracking frame to change size quickly and also some noise that is in the image due to extreme lighting conditions; therefore we decided not to use this method.

FIGURE3.8:HAND TRACKING OF GESTURE IN DIFFERENT LIGHTING



The other method we used was contour based tracking as mentioned in section 3.1.3.2, we start by blurring the image to reduce noise, then using OpenCV we calculate the contours in the image, [67]we then calculate area of each contour and pick the biggest one as it should represent our hand, because our hand occupies the major part of the frame.After this we draw a bounding rectangle around the contour as shown below.

```
gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
blur = cv2.GaussianBlur(gray,(5,5),0)
ret,thresh1 = cv2.threshold(blur,40,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)
contours, hierarchy = cv2.findContours(thresh1,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
max_area = 0
for i in range(len(contours)):
    cnt=contours[i]
    area = cv2.contourArea(cnt)
    #print area
    if area > max_area:
        max_area = area
        ci = i
cnt = contours[ci]
x,y,w,h = cv2.boundingRect(cnt)
cv2.rectangle(originalframe,(x,y),(x+w,y+h),(0,255,0),2)
```

The result of this technique for tracking can be seen in figure 3.9 and 3.10, the green window is the tracking window .Since the results are satisfactory in different lighting conditions , therefore we use this method in our software.

FIGURE3.9 :HAND TRACKING OF GESTURE IN SAME LIGHTING

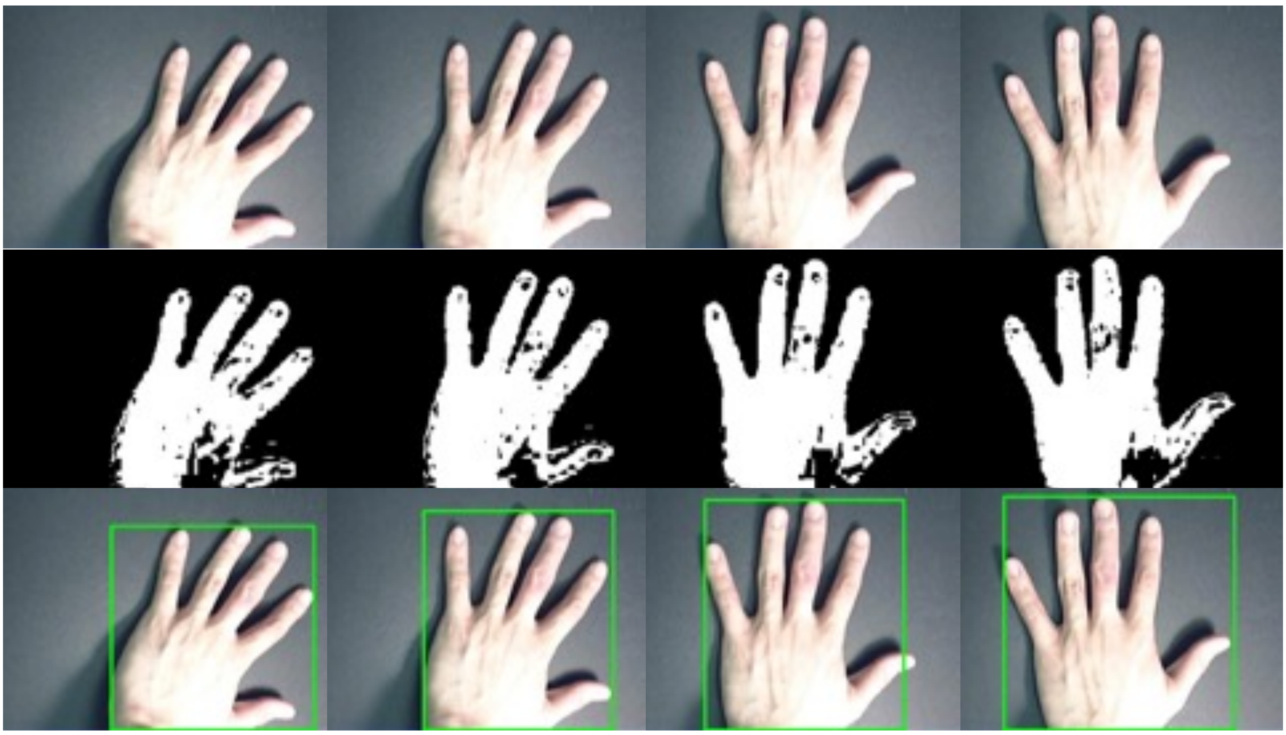


FIGURE3.10:HAND TRACKING OF GESTURES IN DIFFERENT LIGHTING

3.2.3 Feature Extraction

After detection and tracking the next phase is to calculate features and store them to use in the recognition phase.As mentioned in section 3.1.3.3, we decided to use Hu Invariant Moments and Orientation as features. After tracking the hand we calculate these features in the tracking window, we use OpenCV to calculate the moment of the tracking window and then calculate the Hu Moments [68] , at the same time we calculate the orientation of the centre of the tracking window as explained in section 2.3.4.3 and as shown below.

```
x,y,w,h = cv2.boundingRect(cnt)
cv2.rectangle(originalframe,(x,y),(x+w,y+h),(0,255,0),2)
m  = cv2.moments(blur[y:y+h,x:x+w])
centre = [x+w/2,y+h/2]
centerPoints[index] = centre
if index < (len(gestureFrames) - 1):
    centreAngle[index] = math.atan2(centerPoints[index+1,1] - centerPoints[index,1],\
                                    centerPoints[index+1,0] - centerPoints[index,0])
if index == (len(gestureFrames) - 1):
    centreAngle[index] = centreAngle[index - 1]
huInvariants = cv2.HuMoments(m)
huInvariants = vstack((huInvariants,centreAngle[index]))

huInvariantsT = huInvariants.transpose()
featureArray[index] = huInvariantsT
if index == (len(gestureFrames) - 1):
    np.save(gesturePath + 'Results/features2',featureArray)
```

After this we store these features as a vector for each image and stack them to create a matrix for an image sequence.Thus , we get a matrix of 8 columns (7 columns of hu invariant and the last column having tan of the angle as explained in section 2.3.4.3).We use these features for recognition but we don't use all of these features as not all of them are independent of each other as discussed in section 2.3.4.2.

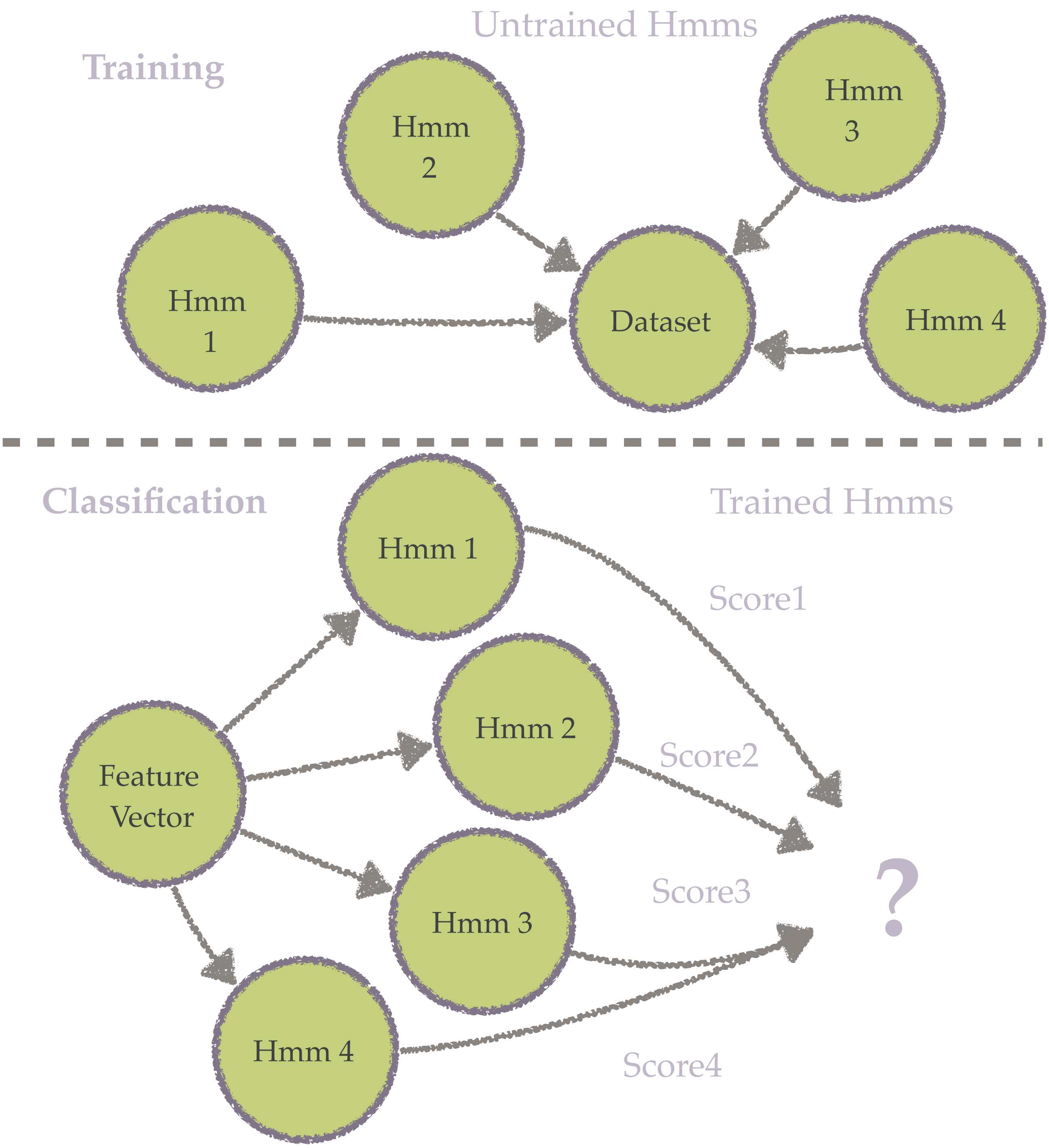
3.2.4 Recognition

After detection and tracking , and feature extraction the next step is to use these features to recognise and classify the hand gestures.As mentioned in section 2.3.5 and section 3.1.3.4 we have decided to use Hidden Markov Models.To implement Hmms in python we have used the sklearn.hmm module version 0.16.1[62] As mentioned in section 3.2.3 after extraction we store all the features in a matrix, we use this matrix or columns of this matrix as a sequence of observable sequence , we use the matrices from all of the image sequences of a certain class of the dataset as an observable sequence to train the hmm for that particular class, we do the same for all the gesture classes we want to recognise.Since the emissions for our case are continuous therefore we use gaussian Hmms in sklearn.hmm. For the learning phase we use an expectation maximisation algorithm generally called baum welch algorithm as shown below.

```
    featureArray = vstack((featureArray,aData))
#print featureArray.shape
#make a hmm model
Model = GaussianHMM(2, covariance_type="full", n_iter=100000)
print("fitting to HMM 1...")
featureArray = featureArray[1:]
#train the Hmm model on amtrix cotianing observable
# sequences aka the features of a particular class
Model.fit([featureArray])
```

After training, we have trained Hmm model, now when we perform a gesture or use image sequence of a gesture , we perform the detection and tracking, and feature extraction and obtain a matrix of features from the image sequence which is our set of observable sequences . Now , in the classification phase we use Viterbi algorithm to calculate the log probability of the new feature matrix which is our set of observable sequence belonging to one of the trained Hmm , then based on the probability score we classify it as the Hmm which got the highest probability. Figure 3.11 shows the two phases.

FIGURE3.11:TRAINING AND CLASSIFICATION PHASE FOR HAND GESTURE RECOGNITION



Chapter 4

4. Evaluation

This section discusses the software system that has been produced, furthermore it discusses the various types of testing that has been done on the system.

4.1. Software Artefact

Based on the various methods and techniques mentioned above , the software system was developed with a simple GUI using python as seen in figure 4.1. On clicking the track button we get a tracking window showing real time hand tracking as shown in figure 4.2, to record a gesture we press ‘r'on the keyboard and to quit we press ‘q’.After this , to recognise we press the Recognise button and the system gives us the name of the gesture and a mock sample of it as shown in fig 4.3.

FIGURE4.1:GUI FOR THE HAND GESTURE RECOGNITION SYSTEM

As you can see in figure 4.2 we draw a hull and a bounding rectangle around the hand in tracking window to show real time tracking.

FIGURE4.2:TRACKING WINDOW FOR THE HAND GESTURE RECOGNITION SYSTEM

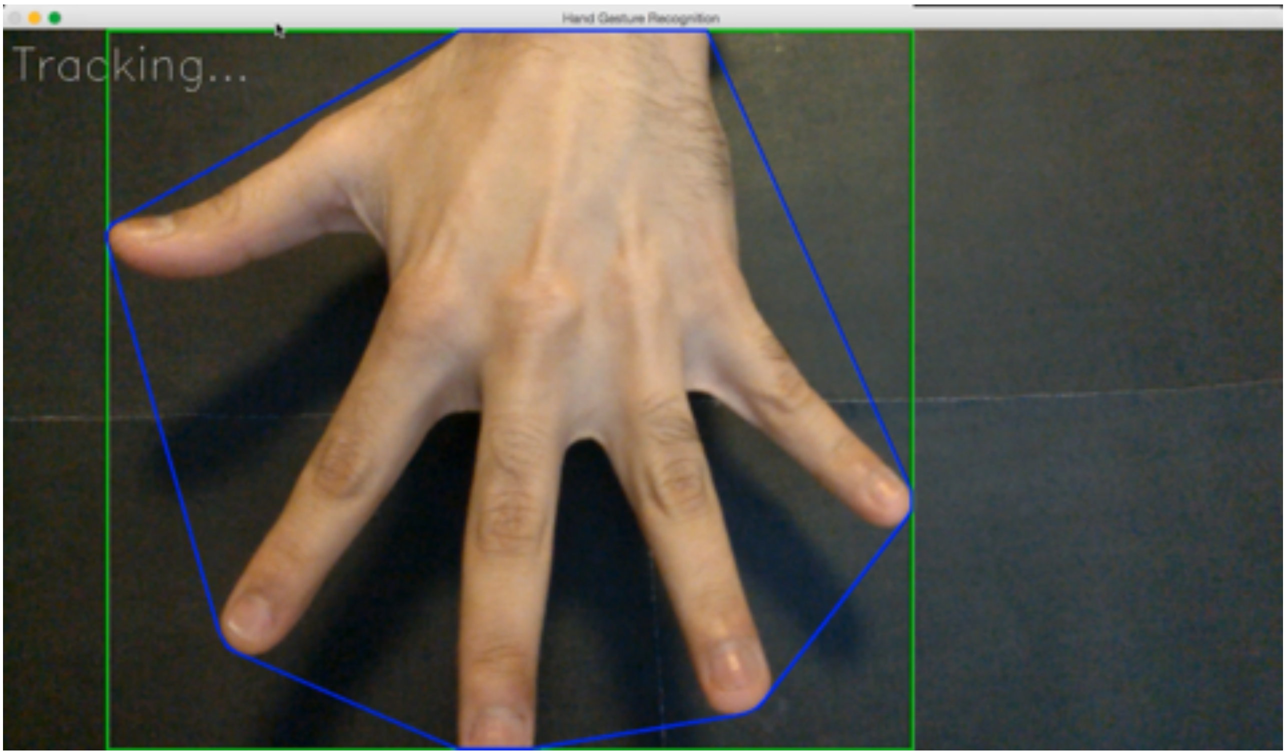
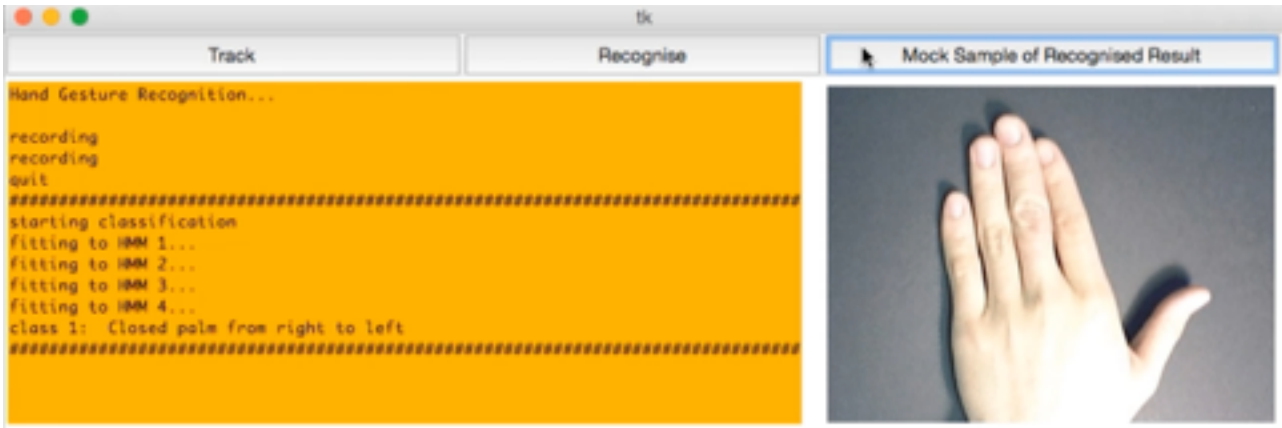


FIGURE4.3:HAND GESTURE RECOGNITION SYSTEM RECOGNISING A GESTURE



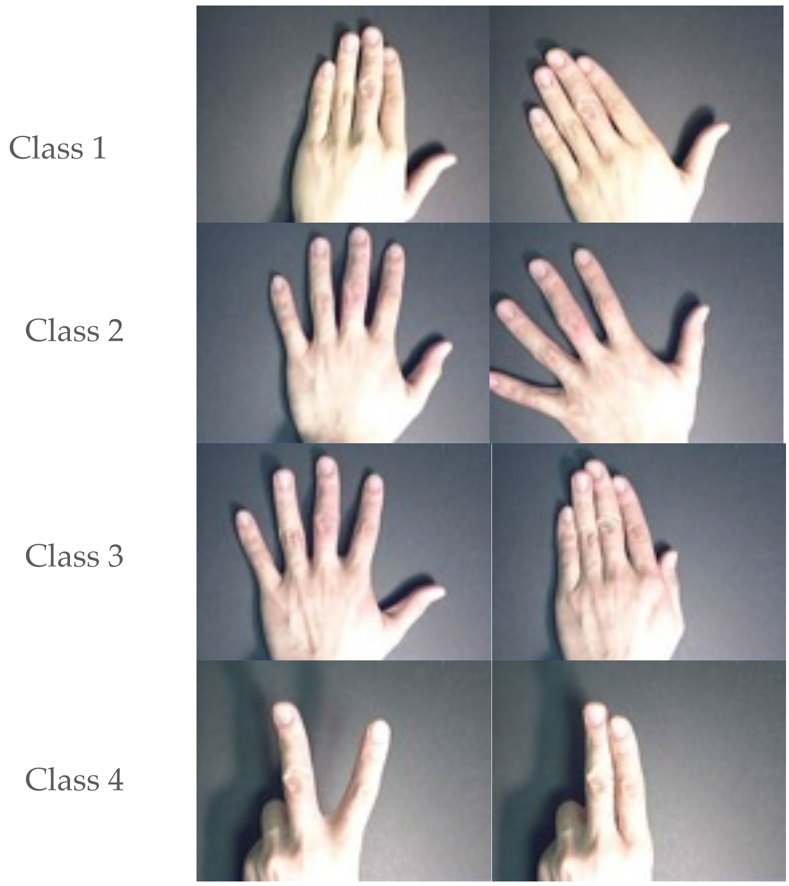
4.2 Testing Results

In order to test the robustness and accuracy of the hand gesture recognition system produced in the project, we use two type of testing; real life testing and testing using the dataset.

4.2.1 System Testing Using Dataset

In order to use the dataset to perform system testing, we use the K-fold cross validation technique.We have 100 image sequences for each of the class in the dataset, we perform detection and tracking methods and feature extraction for all the image sequences and then use these features or a subset of them to train the system and recognise the hand gestures. We have a feature matrix for each of these image sequence , we divide these 100 matrices in to 5 random set each containing 20 matrices, we then train the Hmm on 4 sets of matrices of a class and then we have 1 set of matrices for each class, we combine these sets and randomly create new sets of matrices each containing 20 matrices and then randomly pick one of these set to classify, we perform this multiple times by choosing random sets every time.

FIGURE4.4:HAND GESTURE CLASSES USED IN THE PROJECT



We do this for all the four gesture classes and get results as shown in table 4.1. We define the four classes as shown in figure 4.4 above.

As seen in Table 4.1 we obtain high to satisfactory accuracy for each of the class. After this testing we test for real life performance.

TABLE 4.1: ACCURACY PERCENTAGE OF 5-FOLD CROSS VALIDATION

Iteration	Class 1	Class 2	Class 3	Class 4
1	100%	85%	90%	75%
2	95%	90%	70%	80%
3	90%	75%	85%	80%
4	100%	80%	75%	85%
5	95%	95%	85%	90%

4.2.2 Real Life Testing

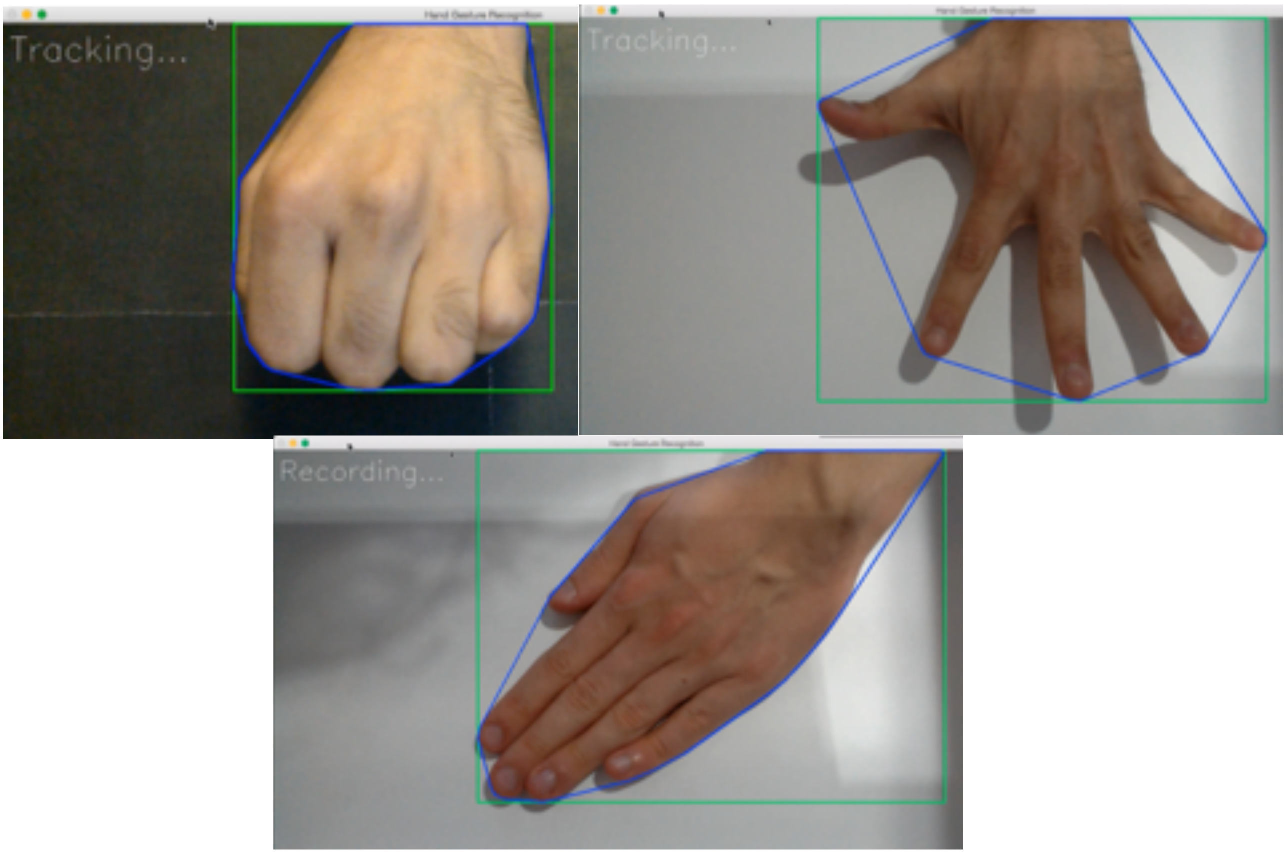
In order to test the system for robustness in detecting and tracking different skin colours in different lighting condition , with different backgrounds and shadows ,and then recognising the various gestures we did various real life test as shown in figure 4.5.

We also performed 50 gestures of each class in two different lighting condition, with different background and with 2 different subjects and calculated the accuracy of classification , the result can be seen in table 4.2 .

TABLE 4.2: ACCURACY PERCENTAGE OF CLASSIFICATION IN REAL LIFE USAGE

Class 1	Class 2	Class 3	Class 4
93%	83%	91%	88%

FIGURE 4.5:DETECTION AND TRACKING BY SYSTEM IN VARIED SITUATIONS



As seen in table 4.2 we get a high percentage of accuracy of classification , a note here is that detection and tracking techniques were working properly in the conditions used for testing.If we performed testing in conditions where the detection and tracking didn't work properly we would get low accuracy, although the system is robust to different lighting condition, skin colour and different but simple background it has some limitations in extreme situations.

Chapter 5

5. Conclusion

This section discusses the planning and management that went into the project, and the information we learnt in the domain of image processing and machine learning.It also gives a conclusion discussing limitations and future work.

5.1 Reflection

As mentioned in Context section , we researched the various approaches and the methods used for hand gesture recognition . We made some assumptions in the very beginning and stuck with them as they made the project more achievable . We found the dataset and the methods to be used as mentioned in the Development section according to our assumptions, as mentioned in the development section we chose two techniques for each phase except the recognition phase and based on the result we got , we chose one out of them or a combination of both.

In terms of image processing I learnt about the various basic techniques for extracting important information from image and techniques like adaptive thresholding, calculating contours and moment, I also learnt about various colour spaces and their suitability for various applications and in order to implement these and other techniques I learnt about OpenCV and the various function it provides.

In terms of Machine Learning , I learnt about markov chains and first order hidden markov models, although I used an API to implement the Hmm I still learnt a lot about the various problems that Hmms are used to solve and the various algorithm that are employed to solve these problems

5.2 Project Conclusion

Hand Gesture Recognition is very broad and difficult problem , in this project I have tried to implement a system that recognises pre-defined hand gesture with some major assumptions like simple background , camera focused on hand region and reduced degree of motion, which is not the case in real life. Therefore, this project aims at only a tiny subset of the larger problem of hand gesture recognition. The software system that has been developed performs well in the conditions which satisfy these assumptions and is able to track and detect hands in real time with almost no latency. Recognition on the other hand requires user interference to record the gesture and then ask the system to classify it. Apart from this , the system is robust to varying non-extreme light conditions, different skin colours, different but simple background, different speeds of motion and different orientation of hand while performing gestures.

5.2.1 Limitation

As discussed above few assumption were made for the project , but the system has few limitations inspite of that, like its inability to detect and track hand if the background is very similar to the skin colour, to detect and track hand in extreme light conditions .Moreover, the recognition phase requires user interference as mentioned above.

5.2.2 Future Work

We can expand on the system in the future to improve detection and tracking to overcome the limitations, for example apart from using just skin colour for detection , we can use some other properties , similarly we can use other techniques for tracking and especially we can work on making the recognition phase more autonomous and recognise the gestures in real time as well.

References

[1] Elmezain M., Al-Hamadi A., Appenrodt J., Michaelis B., "A Hidden Markov Model-based continuous gesture recognition system for hand motion trajectory," Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, vol., no., pp.1-4, 8-11 Dec. 2008. [2] T. Starner, A. Pentland, Visual recognition of American sign language using hidden Markov model, Proceedings of the International Workshop on Automatic Face-and Gesture- Recognition, Zurich, Switzerland, June 1995,pp.189-194.

[2] Gesture - Wikipedia, the free encyclopedia. 2015. Gesture - Wikipedia, the free encyclopedia. [ONLINE] Available at: http://en.wikipedia.org/wiki/Gesture. [Accessed 19 April 2015].

[3] Vision based hand gesture recognition for human computer interaction: a survey - Springer. 2015. Vision based hand gesture recognition for human computer interaction: a survey - Springer. [ONLINE] Available at: http://link.springer.com/article/10.1007%2Fs10462-012-9356-9. [Accessed 19 April 2015].

[] Karam M (2006) A framework for research and design of gesture-based human computer interactions. PhD Thesis, University of Southampton.

[4]Reinhard Klette, 2014. Concise Computer Vision (Undergraduate Topics in Computer Science). 1 Edition. Springer London. ISBN 978-1-4471-6320-6.

[5]Linda G. Shapiro, 2001. Computer Vision. 1 Edition. Prentice Hall.. ISBN 0-13-030796-3.

[6] Tim Morris, 2004. Computer Vision and Image Processing (Cornerstones of Computing). First Edition, First Printing Edition. Palgrave Macmillan Ltd..

[7] Machine learning | artificial intelligence | Encyclopedia Britannica. 2015. machine learning | artificial intelligence | Encyclopedia Britannica. [ONLINE] Available at: http://www.britannica.com/EBchecked/topic/1116194/machine-learning. [Accessed 23 April 2015].

[8] Wernick, Yang, Brankov, Yourganov and Strother, Machine Learning in Medical Imaging, IEEE Signal Processing Magazine, vol. 27, no. 4, July 2010, pp. 25-38

[9] Kanniche MB (2009) Gesture recognition from video sequences. PhD Thesis, University of Nice

[10] A Study on Hand Gesture Recognition Technique 2015. . [ONLINE] Available at: http://ethesis.nitrkl.ac.in/2887/1/209ec1111.pdf. [Accessed 25 April 2015].

[11] Razor3d 2015. . [ONLINE] Available at: http://www.razor3d.co.kr/dlp/Animazoo%20IGS.files/image001.jpg. [Accessed 25 April 2015].

[12] Webel S, Keil J, Zoellner M (2008) Multi-touch gestural interaction in x3d using hidden markov models. In: VRST '08: proceedings of the 2008 ACM symposium on vVirtual reality software and technology. ACM, New York, pp 263–264

[13] Lee, Kevin (2015-04-06). "Leap Motion bounding to more HP desktops, all-in-ones with new keyboard". Techradar. Retrieved 2015-04-08. "Users will be able to access Airspace, Leap Motion's own app store, for a (hopefully) expanding number of applications."

[14] Mitra S, Acharya T (2007) Gesture recognition: a survey. IEEE Trans Syst Man Cybern (SMC)Part C Appl Rev 37(3):311–324

[15] SenzTech. 2015. . [ONLINE] Available at: http://www.senztech.cc/ckfinder/userfiles/images/prod_virtualhandmotion.gif. [Accessed 25 April 2015].

[16] IDS Imaging. 2015. . [ONLINE] Available at: https://en.ids-imaging.com/store/media/catalog/product/cache/2/image/795x795/9df78eab33525d08d6e5fb8d27136e95/c/a/camera-usb2-ensenso-3d-1.jpg. [Accessed 25 April 2015].

[17] Nishikawa A, Hosoi T, Koara K, Negoro D, Hikita A, Asano S, Kakutani H, Miyazaki F, Sekimoto M, Yasui M, Miyake Y, Takiguchi S, Monden M (2003) FAce MOUSe: a novel human-machine interface for controlling the position of a laparoscope. IEEE Trans Robotics Autom 19(5):825–841

[18] David McNeill, 1992. Hand and Mind: What Gestures Reveal about Thought. 1 Edition. University Of Chicago Press.

[19] Boulay B (2007) Human posture recognition for behavior understanding. PhD thesis, Universit'e de Nice- Sophia Antipolis

[20] Bretzner L, Laptev I, Lindeberg T (2002) Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. In: Fifth IEEE international conference on automatic face and gesture recognition, pp 405–410.

[21] Birdal A, Hassanpour R (2008) Region based hand gesture recognition. In: 16th International Conference in central Europe on computer graphics, visualisation and computer vision, pp 1–7

[22] Ju SX, Black MJ, Minneman S, Kimber D (1997) Analysis of gesture and action in technical talks for video indexing, Technical report, American Association for Artificial Intelligence. AAAI Technical Report SS-97-03

[23] Luo Q, Kong X, Zeng G, Fan J (2008) Human action detection via boosted local motion histograms. Mach Vis Appl.

[24] 3-D shape Browser. 2015. . [ONLINE] Available at: http://www.3dcadbrowser.com/th/1/42/42333.jpg. [Accessed 26 April 2015].

[25] Terrillon J, Shirazi M, Fukamachi H, Akamatsu S (2000) Comparative performance of different skin chromi- nance models and chrominance spaces for the automatic detection of human faces in color images. In: Proceedings of the international conference on automatic face and gesture recognition (FG), pp 54–61

[26] Bradski G (1998) Real time face and object tracking as a component of a perceptual user interface. In: IEEE workshop on applications of computer vision. Los Alamitos, California, pp 214–219

[27] Francois R, Medioni G (1999) Adaptive color background modeling for real-time segmentation of video streams. In: International conference on imaging science,

systems, and technology, Las Vegas, pp 227– 232

[28] Saxe D, Foulds R (1996) Toward robust skin identification in video images. In: IEEE international conference on automatic face and gesture recognition, pp 379–384

[29] Yang J, Lu W, Waibel A (1998b) Skin-color modeling and adaptation. In: ACCV, pp 687–694

[30] Sigal L, Sclaroff S, Athitsos V (2004) Skin color-based video segmentation under time-varying illumination. IEEE Trans Pattern Anal Mach Intell 26(7):862–877

[31] Cui Y, Weng J (1996) Hand sign recognition from intensity image sequences with complex background. In: Proceedings of the IEEE computer vision and pattern recognition (CVPR), pp 88–93

[32] Martin J, Devin V, Crowley J (1998) Active hand tracking. In: IEEE conference on automatic face and gesture recognition, Nara, Japan, pp 573–578

[33]Belongie S, Malik J, Puzicha J (2002) Shape matching and object recognition using shape contexts. IEEE Trans Pattern Anal Mach Intell 24(4):509–522

[34] Yin M, Xie X (2003) Estimation of the fundamental matrix from uncalibrated stereo hand images for 3D hand gesture recognition. Pattern Recogn 36(3):567–584

[35] Crowley J, Berard F, Coutaz J (1995) Finger tracking as an input device for augmented reality. In: International workshop on gesture and face recognition, Zurich

[36] Darrell T, Essa I, Pentland A (1996) Task-specific gesture analysis in real-time using interpolated views. IEEE Trans Pattern Anal Mach Intell 18(12):1236–1242

[37] Argyros A, Lourakis MIA (2004a) Real-time tracking of multiple skin-colored objects with a possibly moving camera. In: Proceedings of the European conference on computer vision, Prague, pp 368–379

[39] Cootes TF, Taylor CJ, Cooper DH, Graham J (1995) Active shape models—their training and applications. Comput Vis Image Underst 61(1):38–59

[40] Cheng, Yizong (August 1995). "Mean Shift, Mode Seeking, and Clustering". IEEE Transactions on Pattern Analysis and Machine Intelligence (IEEE) 17 (8): 790–799. doi: 10.1109/34.400568

[41] Yilmaz JA, Javed O, Shah M (2006) Object tracking: a survey. ACM Comput Surv 38:13

[42] Bradski G, Kaehler A (2008) Learning OpenCV, O'Reilly, pp 337–341

[43] Xiangyu W, Xiujuan L (2010) The study of moving target tracking based on Kalman CamShift in the video. In: 2nd International conference on information science and engineering (ICISE), pp 1–4

[44]OpenCV: Meanshift and Camshift. 2015. OpenCV: Meanshift and Camshift. [ONLINE] Available at: http://docs.opencv.org/master/db/df8/tutorial_py_meanshift.html. [Accessed 26 April 2015].

[45] Feng-Sheng Chen, Chih-Ming Fu, Chung-Lin Huang, Hand gesture recognition using a real-time tracking method and hidden Markov models, Image and Vision Computing, Volume 21, Issue 8, 1 August 2003, Pages 745-758, ISSN 0262-8856, http://dx.doi.org/10.1016/S0262-8856(03)00070-2.(http://www.sciencedirect.com/science/article/pii/S0262885603000702)

[46] Image moment - Wikipedia, the free encyclopedia. 2015. Image moment - Wikipedia, the free encyclopedia. [ONLINE] Available at: http://en.wikipedia.org/wiki/Image_moment. [Accessed 26 April 2015].

[47] M. K. Hu, "Visual Pattern Recognition by Moment Invariants", IRE Trans. Info. Theory, vol. IT-8, pp.179–187, 1962

[48] J. Flusser and T. Suk, "Rotation Moment Invariants for Recognition of Symmetric Objects", IEEE Trans. Image Proc., vol. 15, pp. 3784–3790, 2006.

[49] H. Lee and J. Kim.An hmm-based threshold model approach for gesture recognition.IEEE Transactions Pattern Analysis and Machine Intelligence, (pp. 962-976), Oct 1999.

[50] Markov chain - Wikipedia, the free encyclopedia. 2015. Markov chain - Wikipedia, the free encyclopedia. [ONLINE] Available at: http://en.wikipedia.org/wiki/Markov_chain. [Accessed 27 April 2015].

[51] Hidden Markov Models Fundamentals. 2015. . [ONLINE] Available at: http://see.stanford.edu/materials/aimlcs229/cs229-hmm.pdf. [Accessed 27 April 2015].

[52] (ML 14.5) Hidden Markov models (HMMs) (part 2) - YouTube. 2015. (ML 14.5) Hidden Markov models (HMMs) (part 2) - YouTube. [ONLINE] Available at: https://www.youtube.com/watch?v=M_IIW0VYMEA&index=99&list=PLD0F06AA0D2E8FFBA. [Accessed 27 April 2015].

[53] Charniak E (1993) Statistical language learning. MIT Press, Cambridge

[54] 1.2 Isolated Word Recognition. 2015. 1.2 Isolated Word Recognition. [ONLINE] Available at: http://izanami.tl.fukuoka-u.ac.jp/SLPL/HMM/HTKBook/node5.html. [Accessed 27 April 2015].

[55] Andrea C (2001) Dynamic time warping for offline recognition of a small gesture vocabulary. In: Proceedings of the IEEE ICCV workshop on recognition, analysis, and tracking of faces and gestures in real-time systems, July–August, p 83

[56] Senin P (2008) Dynamic time warping algorithm review, technical report. http://csdl.ics.hawaii.edu/ techreports/08-04/08-04.pdf

[57] Dynamic Time Warping. 2015. . [ONLINE] Available at: http://2.bp.blogspot.com/-6OzEsZtrJ3w/UJKbp6wFlFI/AAAAAAAAAOg/K6IZ-z6aGAg/s1600/dynamic_time_warping.png. [Accessed 27 April 2015].

[58] A Survey of Datasets for Human Gesture Recognition . 2015. . [ONLINE] Available at: http://download-v2.springer.com/static/pdf/540/chp%253A10.1007%252F978-3-319-07230-2_33.pdf?token2=exp=1430132003~acl=%2Fstatic%2Fpdf%2F540%2Fchp%25253A10.1007%25252F978-3-319-07230-2_33.pdf*~hmac=89536f3f84e2d2a2f58993449b19a3d4d27b96422ce9e7a9e332dc00e776e908. [Accessed 27 April 2015].

[59] Imperial Computer Vision and Learning Lab. 2015. Imperial Computer Vision and Learning Lab. [ONLINE] Available at: http://www.iis.ee.ic.ac.uk/icvl/ges_db.htm. [Accessed 27 April 2015].

[60] OpenCV | OpenCV. 2015. OpenCV | OpenCV. [ONLINE] Available at: http://opencv.org. [Accessed 27 April 2015].

[61] Python vs Matlab — Pyzo - Python to the people. 2015. Python vs Matlab — Pyzo - Python to the people. [ONLINE] Available at: http://www.pyzo.org/python_vs_matlab.html. [Accessed 27 April 2015].

[62] Hidden Markov Models — scikit-learn 0.16.1 documentation. 2015. Hidden Markov Models — scikit-learn 0.16.1 documentation. [ONLINE] Available at: http://scikit-learn.org/stable/modules/hmm.html. [Accessed 27 April 2015].

[63] CIELab colour space . 2015. . [ONLINE] Available at: http://www.aces.edu/dept/fisheries/education/pond_to_plate/documents/ExplanationoftheLABColorSpace.pdf. [Accessed 27 April 2015].

[64]Color Spaces for skin detection . 2015. . [ONLINE] Available at: https://www.inf.pucrs.br/~pinho/CG/Trabalhos/DetectaPele/Artigos/OPTIMUM%20COLOR%20SPACES%20FOR%20SKIN%20DETECTION.pdf. [Accessed 27 April 2015].

[65] Miscellaneous Image Transformations — OpenCV 2.4.11.0 documentation. 2015. Miscellaneous Image Transformations — OpenCV 2.4.11.0 documentation. [ONLINE] Available at: http://docs.opencv.org/modules/imgproc/doc/miscellaneous_transformations.html. [Accessed 27 April 2015].

[66] Meanshift and Camshift — OpenCV-Python Tutorials 1 documentation. 2015. Meanshift and Camshift — OpenCV-Python Tutorials 1 documentation. [ONLINE] Available at: http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_video/py_meanshift/py_meanshift.html. [Accessed 27 April 2015].

[67] Hand Tracking And Gesture Detection (OpenCV) | Lakshminarayanan Srinivasan. 2015. Hand Tracking And Gesture Detection (OpenCV) | Lakshminarayanan

Srinivasan. [ONLINE] Available at: http://s-ln.in/2013/04/18/hand-tracking-and-gesture-detection-opencv/. [Accessed 27 April 2015].

[68] Structural Analysis and Shape Descriptors — OpenCV 2.4.11.0 documentation. 2015. Structural Analysis and Shape Descriptors — OpenCV 2.4.11.0 documentation. [ONLINE] Available at: http://docs.opencv.org/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html. [Accessed 27 April 2015].