

Acknowledgment

I take the opportunity to express my deepest gratitude and appreciation to all those who have helped me directly or indirectly towards the successful completion of this project.

I wish to place on records our ardent and earnest gratitude to Principal Dr. S. Ayoob, my project coordinator and guide Dr. Ansamma John (Head of Department, Computer Science and Engineering), Prof Reena Mary George and my senior advisor Prof. Shameem Ansar. Their tutelage and guidance was the leading factor in translating my effort to fruition. Their prudent and perfective vision has shown light on my trail to triumph. They have played a major role in providing with all facilities for the completion of this work.

I would like to take this opportunity to express our thanks towards the teaching and non-teaching staff in the Department of Computer Science and Engineering, TKMCE, Kollam, for their valuable help and support. I also thank all my family, friends and well-wishers who greatly helped in my endeavour.

Contents

1	Introduction	1
2	Literature Review	4
2.1	Gesture Recognition Challenges and Approaches	5
2.1.1	Sensor Based	5
2.1.2	Vision Based	6
2.2	Automatic Speech Recognition System	13
2.2.1	CMU SPHINX	14
2.2.2	GOOGLE API	14
2.2.3	MICROSOFT API	15
3	Design and Implementation	16
3.1	Gesture Recognition Module	16
3.1.1	Dataset	16
3.1.2	Architecture	17
3.1.3	Preprocessing	18
3.1.4	Gesture Identification	19
3.2	Mouse Control Module	22
3.2.1	Preprocessing	22
3.2.2	Mouse Events	23
3.3	Voice Control Module	25
3.3.1	Speech Recognition	25

4	Results and Discussions	28
4.1	Skin Segmentation	28
4.2	Mouse Movement Analysis	34
4.3	Voice Recognition Analysis	35
5	Conclusion	37
	Bibliography	38

List of Figures

3.1	CNN Architecture	18
3.2	Mouse Control Flow Graph	24
3.3	Mouse Control Flow Graph	24
3.4	Speech Recognition Process	27
4.1	Input video frame frame without a hand inside designated green window.	29
4.2	Output video frame without a hand inside designated green window. . . .	29
4.3	Input video frame with a hand inside designated green window	30
4.4	Output video frame with a hand inside designated green window	30
4.5	Input video frame with a hand inside designated green window	31
4.6	Output video frame with a hand inside designated green window	31
4.7	Input video frame with a hand inside designated green window using background subtraction method	32
4.8	Output video frame with a hand inside designated green window using background subtraction method	32
4.9	Defects of 1 is detected when we use angle measures only	34
4.10	Defects of 0 is detected when we use angle measures and distance mea- sures	35

List of Tables

3.1	Gesture and Its Actions	21
3.2	Number of Defects and Its Operations	23
3.3	Voice Commands and their Operations	27
4.1	Gesture Prediction Result	33
4.2	File and the Text on each file	35
4.3	Result of Google API	36

Abstract

An increasingly impactful part of everyday life is how we interact with our technological devices most notably our computers. Now research focuses only on improving current mainstream devices out there today. Only a few modes of Human Computer interaction exist today: namely through keyboards, mice, touch screens, and other devices. Each of these devices had with their own limitations when adapting to more powerful and versatile hardware in computers. There are ways we could build a simple gesture control mode of human control interaction without the need to necessarily build another device. When tackling this problem of creating a new mode of human computer interaction we knew we could utilize a combination of built in functions that is typically provided in most computer designs. Specifically we have exploited the built in webcam that has become a standard feature in most computers. This feature provides us a way to track and respond user hand free movements and gestures. Using a combination object detection and recognition, the following project successfully builds a computationally inexpensive static hand gesture recognition system using a simple RGB webcam creating a truly more natural form of human computer interaction.

Chapter 1

Introduction

Computer technology has tremendously grown over the past decade and has become a necessary part of everyday live. The primary computer accessory for Human Computer Interaction is the mouse. The mouse is not suitable for Human Computer Interaction in some real life situations, such as with Human Robot Interaction. There have been many researches on alternative methods to the computer mouse for Human Computer Interaction. The most natural and intuitive technique for Human Computer Interaction, that is a viable replacement for the computer mouse is with the use of hand gestures. This project is therefore aimed at investigating and developing a Computer Control system using hand gestures.

Most laptops today are equipped with webcams, which have recently been used in security applications utilizing face recognition. In order to harness the full potential of a webcam, it can be used for vision based Computer Control, which would effectively eliminate the need for a computer mouse or mouse pad. The usefulness of a webcam can also be greatly extended to other Human Computer Interaction application such as a sign language database or motion controller. Over the past decades there have been significant advancements in Human Computer Interaction technologies for gaming purposes, such as the Microsoft Kinect and Nintendo Wii. These gaming technologies provide a more natural and interactive means of playing videogames. Human

Computer Interaction using hand gestures is very intuitive and effective for one to one interaction with computers and it provides a Natural User Interface. There has been extensive research towards novel devices and techniques for cursor control using hand gestures. Besides Human Computer Interaction, hand gesture recognition is also used in sign language recognition, which makes hand gesture recognition even more significant.

Generally, Hand Gesture Recognition technology is implemented using “Data Gloves” which in turn leads to additional cost. Also, using additional devices, involves more amount of maintenance. Webcam is an easily available device and In our project, we will implement a hand gesture recognizer which is capable of detecting a hand gesture in webcam frames. In future, it may be considered that willing to be more natural and more comforted, human being, who has been communicated with computers through mouse, keyboards, several user interfaces and some virtual environments, may use their bare hands to interact with machines without any mediator. As the set of materials above, recognition of hand gestures and postures is a satisfactory way to first steps of solutions instead of using keyboards, mouse or joysticks. A very common disease known as Parkinson’s disease is very relevant in now a days. This disease is caused due to excessive use of keyboard and mouse. Use of Gesture recognition technology will prevent this in future.

Our proposed Approach is vision based, In this we divide the entire system into 3 modules. That is a Gesture Recognition Module, In this we train the sytem with 15 gestures using a Convolutional Neural Network and when we input the gesture through webcam the system identifies the gesture and corresponding class is predicted there by produce the corresponding functions or operations such as open web browser, shutdown system etc. The second module is the mouse control module in this the movement of hand is recorded from the webcam and the mouse pointer is moved according to that. And also by identifying the number of fingertips the mouse actions such as right click, left click are performed. The final module is a voice control module which makes the user more comfortable such as by inputing voice through the mi-

crophone user can open various applications. The aim of this project is to reduce the use of mouse and keyboard and also make the computer interaction more easily and user-friendly.

Chapter 2

Literature Review

The aim of building hand gesture recognition system is to create a natural interaction between human and computer where the recognized gestures can be used for controlling a computer or conveying meaningful information. How to form the resulted hand gestures to be understood and well interpreted by the computer considered as the problem of gesture interaction. Human computer interaction refers to the relation between the human and the computer, Gestures are used for communicating between human and machines. Gestures can be static which require less computational complexity or dynamic which are more complex but suitable for real time environments. Different methods have been proposed for acquiring information necessary for recognition gestures system. Some methods used additional hardware devices such as data glove devices and colour markers to easily extract comprehensive description of gesture features. Other methods based on the appearance of the hand using the skin colour to segment the hand and extract necessary features, these methods considered easy, natural and less cost comparing with methods mentioned before. This work demonstrates a Gesture Recognition system used to control personal computer which reduces the usage of mouse and keyboard.

2.1 Gesture Recognition Challenges and Approaches

Gestures recognition involves complex processes such as motion modelling, motion analysis, pattern recognition and machine learning. It consists of methods with manual and non-manual parameters. The structure of environment such as background illumination and speed of movement affects the predictive ability. The difference in viewpoints causes the gesture to appear different in 2D space. In some research, signer wears wrist band or coloured glove to aid the hand segmentation process, such as in. The use of coloured gloves reduces the complexity of segmentation process. Several anticipated problems in a dynamic recognition, includes temporal variance, spatial complexity, repeatability and connectivity as well as multiple attributes such as change of orientation and region of gesture carried out. There are several evaluation criteria to measure the performance of a gesture recognition system in overcoming the challenges. These criteria are scalability, robustness, real-time performance and user-independent. There are mainly two type of approaches were used, Vision based and Sensor based approaches so let us look in detail about these two approaches.

2.1.1 Sensor Based

Sensor-based approaches generally relies on the use of sensors which are physically attached to users to collect position, motion and trajectories of fingers and hand data. This approach requires the use of sensors, instruments to capture the motion, position, and velocity of the hand. These approaches reduce the need of pre-processing and segmentation stage, which are essential to vision-based gesture recognition. Features such as flex angle of fingers, orientation and the absolute position of hand are often in 3D space, and hence it contains the depth information which is useful in telling distance of gesture away from source of sensors. Sensor-based approaches often requires users to wear a glove with sensors or with probes attached to the arm of users. These instruments are required to be set up prior to the recognition, and these often limit the approaches to a laboratory setup. Some of the sensor based techniques

are discussed below

1. Data glove

Data gloves used in gesture and recognition utilizes IMU sensors such as gyroscope and accelerometer to obtain the orientation, angular, acceleration information. Flex sensors are present in some data gloves to obtain finger bending information. VLP-Data glove is a pair of flex-sensor gloves that consist of fiber optic transducer, which measures the flex angles, position, and orientation data.

2. WiFi and Radar

Another type of technology used for gesture recognition is WiFi oriented gesture control. The authors claimed that this method is much simple to be applied as compared to Kinect technology. It uses WiSee technology that consists of multiple antennas to focus on one user to detect the user's gesture. Signals used in Wifi do not require line of sight and can traverse through walls. It utilizes the properties of Doppler shift, which is the change in frequency of a wave as its sources move relative to the observer.

2.1.2 Vision Based

Vision based gesture recognition method is based on proper image processing techniques and which is having very less cost compared to Sensor based approach. Vision-based approaches differs from sensor-based approaches mainly by the data-acquisition method. The process of gesture recognition can be categorized into few stages in general, namely data acquisition, pre-processing, segmentation, feature extraction and classification. The input of static gesture recognition is single frames of images, while dynamic sign languages takes video, which is continuous frames of images as input. The methodologies and techniques used by vision-based gesture recognition are discussed below.

1. Data acquisition

In vision-based gesture recognition, the data acquired is frame of images. The input of such system is collected using images capturing devices such as standard video camera, webcam, stereo camera, thermal camera or more advanced active techniques such as Kinect and LMC. Stereo cameras,0 Kinect and LMC are 3D cameras which can collect depth information.

2. Image preprocessing

Image pre-processing stage are performed to modify the image or video inputs to improves the overall performance of the system. Median filter and Gaussian filter are some of the commonly used techniques to reduce noises in images or video acquired. Next, morphological operation is also widely used to remove unwanted information. For instance, first threshold the input image into binary image, then median and Gaussian filters is used to remove noises followed by using morphological operations as the pre-processing stage. In some researches, the images captured are downsized into a smaller resolution prior to subsequent stages. This technique is used to reduce the resolution of the input image is able to improve the computational efficiency. In this research, division by 64 is the optimum scale as it reduced processing time by 43.8% without affecting the overall accuracy. Histogram equalization is used to enhance the contrast of the input images taken under different environment to uniform the brightness and illumination of the images.

3. Segmentation

Segmentation is the process of partitioning images into multiple distinct parts. It is a stage whereby the Region of Interest (ROI), is segmented from the remaining of the image. Segmentation method can be contextual or non-contextual. Contextual segmentation takes the spatial relationship between features into account, such as edge detection techniques. Whereas a non-contextual segmentation does not consider spatial relationship but group pixels based on global

attributes.

(a) Skin color segmentation

Skin color segmentation are mostly performed in RGB, YCbCr, HSV and HSI color spaces. Several challenges toward achieving a robust skin color segmentation is sensitivity to illumination, camera characteristic and skin color. HSV color space is popular as the Hue of palm and arm differs greatly, hence palm can be segmented from the arm easily. In RGB color space, using the rule of $R \geq G \geq B$ and matching with pre-stored sample skin color to find the skin color. It is found that YCbCr is more robust for skin color segmentation compared to HSV in different illumination condition.

(b) Other segmentation method

This method involves applying Continuously Adaptive Mean Shift (CAMShift) in HSV color space to create a histogram of skin pixels to find the suitable segmentation threshold value. Canny edge detection is then applied followed by dilation and erosion. Edge traversal algorithm is used lastly to segment the hand gesture from the background.

While comparing the performance of Sobel edge detection, low pass filtering, histogram equalization, skin color segmentation in HSI color space and desaturation, and found that desaturation provides highest accuracy. Desaturation process includes first converting into grayscale image by removing the chromatic channel while preserving only the intensity channel in HSI color space.

4. Feature extraction

Feature extraction is the transformation of interesting parts of input data into sets of compact feature vectors. In gesture recognition context, the features extracted should contain relevant information from the hand gestures input and represented in a compact version which serves as an identity of the gesture to be

classified apart from other gestures. Some of the feature extraction techniques are discussed here.

(a) Shift invariant feature transform (SIFT)

SIFT is a scale and rotation invariant feature extraction technique. SIFT describe an image by its interest points whereby detection requires multi-scale approach. At each level of the pyramid, the image is rescaled and smoothed by Gaussian function. The scale-space is defined by function, $L(x, y, \sigma)$ given in Eq (2.1)

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (2.1)$$

The key-points extracted are the maxima and minima, which are calculated using difference-of-Gaussian (DoG) function, $D(x, y, \sigma)$. The Gaussian function convolved with the images, $D(x, y, \sigma)$ which is computed by subtracting two subsequent scales which is separated by a constant scale factor k with $k = \sqrt{2}$ as the optimum value.

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (2.2)$$

At each point, $D(x, y, \sigma)$ is compared with eight neighbors of its scale, and nine neighbors up and down one scale. If the $D(x, y, \sigma)$ value is the maximum or minimum among the points, then it is extrema. In key-point localization stage, key-points with low contrast or are poorly localized are removed. The location of extremum, \mathbf{x} is given as

$$\hat{\mathbf{x}} = -\frac{\partial^2 D^{-1}}{\partial x} \frac{\partial D}{\partial x} \quad (2.3)$$

In orientation assignment, each key-points are assigned a consistent orientation based on local image properties. Finally, the SIFT descriptors is cre-

ated in this stage by first lining up the key-points by offsetting the orientation. The matching of SIFT descriptors can then be performed by calculating the nearest neighbor and the ratio of closest-distance to second-closest distance. SIFT is invariant to a certain range of affine transformation, illumination variation, and changes in 3D viewpoint.

(b) Speeded up robust feature (SURF)

SURF is developed based on SIFT. SIFT constructs scale pyramid, convolving the upper and lower scales of the image with DoG operator and searching the local extreme in scale space. Meanwhile, SURF scales filter up instead of iteratively reducing the image size. In SIFT, Laplacian of Gaussian (LoG) is approximated with DoG for finding scale-space. SURF approximates LoG with Box Filter. The convolution of box filter can be calculated easily using integral images, which is a fast and effective method in calculating the sum of pixels value.

In detection of key-points or descriptors, SURF uses an integer approximation of the determinant of Hessian blob detector. Integral image is the sum of intensity value for points in the image with location less than or equal to (x, y) as shown

$$S(x, y) = \sum_{i=1}^x \sum_{j=1}^y I(i, j) \quad (2.4)$$

SURF employs hessian blob detector to obtain interest points. The determinant of Hessian matrix describes the extent of the response. Hessian matrix with point x and scale σ is defined as

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{yx}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad (2.5)$$

where $L_{xx}(x, \sigma)$ is the convolution of the image with the second-order derivative of the Gaussian. To make the system scale-invariant, the scale space is

realized as an image pyramid. With the use of integral image and box filter, the scale space can be realized by up-scaling. Finally, non-maximum suppression is applied in a $3 \times 3 \times 3$ neighborhood to localize interest point in the image. Key-points between two images are matched as nearest neighbours.

(c) Principal Component Analysis (PCA)

PCA is a mathematical operation which utilizes orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of uncorrelated variables called principal components. Given a training set of M images with an S -dimensional vector, PCA finds a t -dimensional subspace which its basis vectors correspond to the maximum variance direction in the original image space.

The dimension of the new subspace is usually lower, where $t \ll s$. The mean, of all images in the training set given in equation, with x_i as the i^{th} image with its columns concatenated in a vector.

$$\mu = \frac{1}{M} \sum_{i=1}^M x_i \quad (2.6)$$

PCA basis vectors are defined as eigenvectors of the Scatter matrix, S_T is computed as

$$S_T = \sum_{i=1}^M (x_i - \mu) \cdot (x_i - \mu)^T \quad (2.7)$$

The eigenvectors and corresponding eigenvalues are calculated and the eigenvectors are stored by decreasing eigenvalues order. The eigenvectors with lower eigenvalues contains less information on the distribution of data, and these are filtered to reduce the dimensionality of data.

5. Classification

Classification can be categorized into supervised and unsupervised machine

learning techniques. Supervised machine learning is a technique that teaches the system to recognize certain pattern of input data, which are then used to predict future data. Supervised machine learning takes in a set of known training data and it is used to infer a function from labeled training data. An unsupervised machine learning is used to draw inferences from datasets with input data with no labeled response. Since no labeled response is fed into the classifier, there is no reward or penalty weightage to which classes the data is supposed to belong.

(a) Artificial Neural Network

ANN is an information-processing system with several performance characteristics in common with that of biological neural networks. ANN is generally defined by three parameters, namely the interconnection pattern between different layers of neurons, the weight of interconnections, and the activation function. A neuron has inputs $x_1, x_2 \dots x_n$, which each are labelled with a weight $w_1, w_2 \dots w_n$ that measures the permeability. The neuron function can be represented as nonlinear weighted sum as shown

$$y = k \sum_{i=1}^n w_i x_i \quad (2.8)$$

where K is the activation function

(b) Support Vector Machine (SVM)

SVM is a supervised machine learning technique. It finds the optimal hyperplane to separate the data points. SVM maximize the margin around the separating hyperplane. Optimization techniques are employed in finding the optimal hyper plane. Two hyperplanes are found which best represent the data. w is the weight vector for w , for training data $(x_1, y_1), \dots, (x_n, y_n)$, where y_i are either -1 or 1, indicating to which class the data x_i belong. The weight vector decides the orientation of decision boundary,

whereas bias point, b decides its location. The hyperplane can be represented as

$$\vec{w} \cdot \vec{x}_i + b = 0 \quad (2.9)$$

The points above the hyperplane will have positive y_i , and points below will have negative y_i . The distance between the support vector and plane is

$$distance = \frac{1}{\|\vec{w}\|} \quad (2.10)$$

The Margin, M is twice the distance to support vector, hence margin is defined as

$$M = \frac{2}{\|\vec{w}\|} \quad (2.11)$$

To maximize the margin, M we need to minimize w

$$minL = \frac{1}{2} \|\vec{w}\|^2 \quad (2.12)$$

where $y_i(\vec{w} \cdot \vec{x} = b) \geq 1$

SVM has better performance over the other methods. SVM with linear kernel perform better than non-linear Gaussian kernel. The method of using SIFT to extract features from images followed by quantization using K-means clustering before mapped into BoF classification using SVM has shown promising results.

2.2 Automatic Speech Recognition System

Automatic Speech Recognition (ASR) is commonly employed in everyday applications. One of the goals of speech recognition is to allow natural communication between humans and computers via speech. There are a number of commercial and open-source systems such as ATT Watson, Microsoft API Speech, Google Speech API,

Amazon Alexa API, Nuance Recognizer, WUW,HTK and Dragon.

Three systems were selected for our evaluation in different environments: Microsoft API, Google API, and Sphinx-4 automatic speech recognition systems.

2.2.1 CMU SPHINX

The Sphinx system has been developed at Carnegie Mellon University (CMU). Currently, CMU Sphinx has a large vocabulary, speaker independent speech recognition codebase, and its code is available for download and use". The Sphinx has several versions and packages for different tasks and applications such as Sphinx-2, Sphinx-3 and Sphinx-4. Also, there are additional packages such as Pocketsphinx, Sphinxbase, Sphinxtrain. In this paper, the Sphinx-4 will be evaluated. The Sphinx-4 has been written by Java programming language. Moreover, its structure has been designed with a high degree of flexibility and modularity". According to Juraj Kačur, "The latest Sphinx-4 is written in JAVA, and Main theoretical improvements are: support for finite grammar called Java Speech API grammar, it doesn't impose the restriction using the same structure for all models". There are three main components in the Sphinx-4 structure, which includes the Frontend, the Decoder and the Linguist. According to Willie Walker and other who have worked in Sphinx-4, "we created a number of differing implementations for each module in the framework. For example, the Frontend implementations support MFCC, PLP, and LPC feature extraction; the Linguist implementations support a variety of language models, including CFGs, FSTs, and N- Grams; and the Decoder supports a variety of Search Manager implementations". Therefore, Sphinx-4 has the most recent version of an HMM-based speech and a strong acoustic model by using HMM model with training large vocabulary.

2.2.2 GOOGLE API

Google has improved its speech recognition by using a new technology in many applications with the Google App such as Goog411, Voice Search on mobile, Voice Ac-

tions, Voice Input (spoken input to keypad), Android Developer APIs, Voice Search on desktop, YouTube transcription and Translate, Navigate, TTS. After Google, has used the new technology that is the deep learning neural networks, Google achieved an 8 percent error rate in 2015 that is reduction of more than 23 percent from year 2013. According to Pichai, senior vice president of Android, Chrome, and Apps at Google, “We have the best investments in machine learning over the past many years. Indeed, Google has acquired several deep learning companies over the years, including DeepMind, DNNresearch, and Jetpac”.

2.2.3 MICROSOFT API

Microsoft has developed the Speech API since 1993, the company hired Xuedong (XD) Huang, Fil Alleva, and Mei-Yuh Hwang “three of the four people responsible for the Carnegie Mellon University Sphinx-II speech recognition system, which achieved fame in the speech world in 1992 due to its unprecedented accuracy. the first Speech API is (SAPI) 1.0 team in 1994” . Microsoft has continued to develop the powerful speech API and has released a series of increasingly powerful speech platforms. The Microsoft team has released the Speech API (SAPI) 5.3 with Windows Vista which was very powerful and useful. On the developer front, ”Windows Vista includes a new WinFX® namespace, System Speech. This allows developers to easily speech-enable Windows Forms applications and apps based on the Windows Presentation Framework”.

Microsoft has focused on increasing emphasis on speech recognition systems and improved the Speech API (SAPI) by using a context- dependent deep neural network hidden Markov model (CD-DNN-HMM). According to the researchers who have worked with Microsoft to improve the Speech API and the CD-DNN-HMM models, they determined that the large-vocabulary speech recognition that achieves substantially better results than a Context-Dependent Gaussian Mixture Model Hidden Markov model..

Chapter 3

Design and Implementation

This chapter will cover the details explanation of methodology that is being used to make this project complete and working well. This final year project used three major steps to implement project starting from planning, implementing and testing. All the methods used for finding and analyzing data regarding the project related. This project mainly consist of 3 modules Gesture Recognition Module, Mouse Control Module and Voice Control Module. Their Implementations are discussed below.

3.1 Gesture Recognition Module

Recent advances in the design of models with deep architectures, especially convolutional networks have paved the way for a vast number of different CNN architectures designed to handle all sorts of data. We created a CNN which look a lot similar to a MNIST classifying model which is used to classify digits using both Tensorflow and Keras.

3.1.1 Dataset

For the purpose of training our model we used the preprocessed American sign language dataset is used. From the dataset we had taken only 15 gestures for our project.

For each gesture 2400 images which were 50x50 pixels size is stored. The Training to Test ratio is 80:20.

3.1.2 Architecture

The initial part was to load all the images into a binary file, That is train images were loaded into one binary file and their labels were stored into another binary file. Similarly test images were loaded into one binary file and their labels were stored into another binary file.

cnn used here consists of convolutional, max pooling, dropout, and flatten layers that use relu and softmax activation functions. An accuracy of 96 percent is obtained after training the model for 20 epochs.

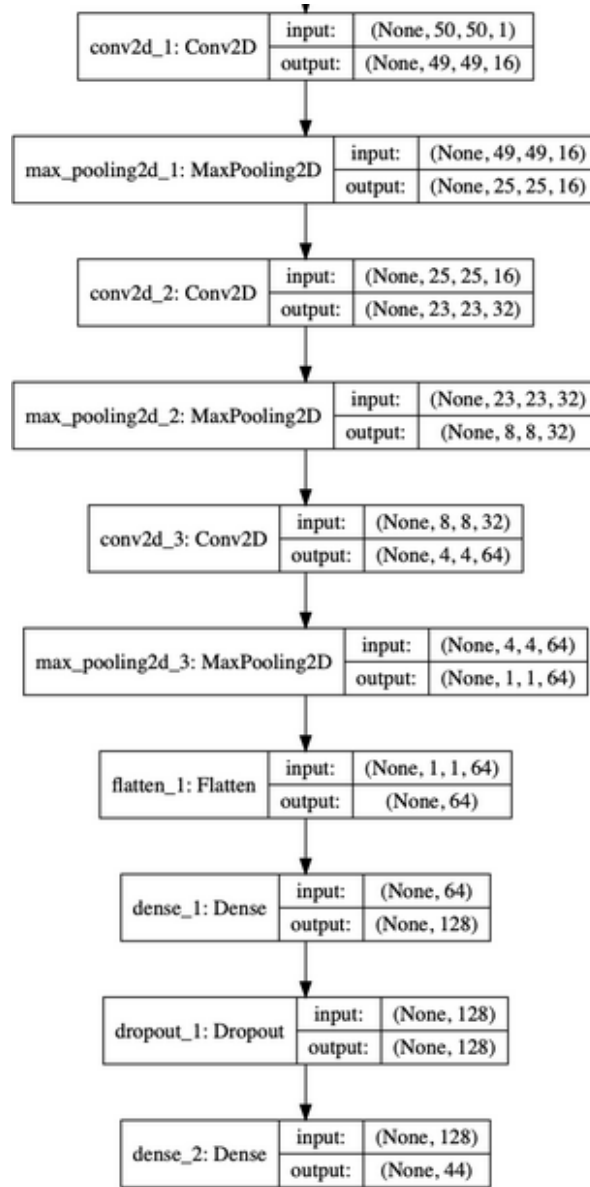


Figure 3.1: CNN Architecture

3.1.3 Preprocessing

In this the input is taken from the camera and the image is resized into 50x50 pixel size. The hand is placed on the area we specified from the specified region that part is taken and it is processed. Skin detection can be defined as detecting the skin colour pixels in an image. Skin detection using colour information is commonly used techniques. But the problem in this method is that when light conditions change the skin

pixel values may varies from the current values which makes the detection more difficult. To solve these problem we use background subtraction method, It is very effective on fixed backgrounds.

So we use Background subtraction method (BS). It is a common and widely used technique for generating a foreground mask (namely, a binary image containing the pixels belonging to moving objects in the scene) by using static cameras. As the name suggests, BS calculates the foreground mask performing a subtraction between the current frame and a background model, containing the static part of the scene or, more in general, everything that can be considered as background given the characteristics of the observed scene.

After obtaining the skin segmented binary image, the next step is to perform edge detection to obtain the hand contour in the image. The OpenCV function FindContours() uses an order finding edge detection method to find the contours in the image. In the contour extraction process, we are interested in extracting the hand contour so that shape analysis can be done on it to determine the hand gesture. It can be seen that besides the hand contour, there are lots of small contours in the image. To isolate the hand contour, we assume that the hand contour is the largest contour and thereby ignoring all the noise contours in the image. After this we get the gesture and it is passed to the trained model.

3.1.4 Gesture Identification

After we get the binary image of the hand we placed on the region it is passed to the model we trained and the model predict which class label it belongs. If the predicted probability is greater than 95% it is identified as a good gesture and corresponding applications are opened using the system calls. There was a problem we identified during this, That is when we place our hand in the region specified the system will take the frames continuously and feed each frame into the CNN so there by opening an application repeatedly. To avoid this we add a special gesture in between. So once

a gesture is identified and then the next frames will not pass into the CNN unless we show that special gesture. Such a way we added 15 gestures and their functionalities. Some of the gestures and their functionalities are given below.





Gesture	Action
	Mute Volume
 <small>www.TeachmeanProblems.net</small>	Open Web Browser
	Open Media Player
 <small>www.TeachmeanProblems.net</small>	Shutdown the System

Table 3.1: Gesture and Its Actions

3.2 Mouse Control Module

The functionality of the cursor was controlled by different hand gestures provided. In order to identify the gestures, the hand must be isolated from the current frame. The method used for isolating the hand is described below.

3.2.1 Preprocessing

The same background subtraction method is used here also. It is used for generating a foreground mask (namely, a binary image containing the pixels belonging to moving objects in the scene) by using static cameras. As the name suggests, BS calculates the foreground mask performing a subtraction between the current frame and a background model, containing the static part of the scene or, more in general, everything that can be considered as background given the characteristics of the observed scene. After obtaining the skin segmented binary image, the next step is to perform edge detection to obtain the hand contour in the image. The OpenCV function `FindContours()` uses an order finding edge detection method to find the contours in the image. In the contour extraction process, we are interested in extracting the hand contour so that shape analysis can be done on it to determine the hand gesture. It can be seen that besides the hand contour, there are lots of small contours in the image. To isolate the hand contour, we assume that the hand contour is the largest contour and thereby ignoring all the noise contours in the image.

3.2.2 Mouse Events

Once the maximum contour is found we will look forward to number of defects so that we can identify how many fingers are detected in the given frame. For identifying defect, we need to find the convex Hull part in the maximum contour. The defects are identified if the angle between two convex Hull part is less than 90° . Once defects are counted, we can say how many fingers are detected in a given frame. The gestures are created according to the number of fingers detected.

Number of Fingertips Detected	Operations Performed
One	Move Cursor
Two	Left Click
Three	Right Click

Table 3.2: Number of Defects and Its Operations

Once the hand gestures are recognized, it will be a simple matter of mapping different hand gestures to specific mouse functions. It turns out that controlling the computer cursor, in the python programming language is relatively easy. By including the pyautogui library into the program, it will allow control of the computer cursor.

The movement of mouse is controlled according to the movement of fingertip. The coordinate of finger tip is calculated by taking the maximum value of x and y of the convex Hull part in the current contour. Once the value of x and y are found, by using pyautogui library we can map the movement of cursor according to the values of x and y.

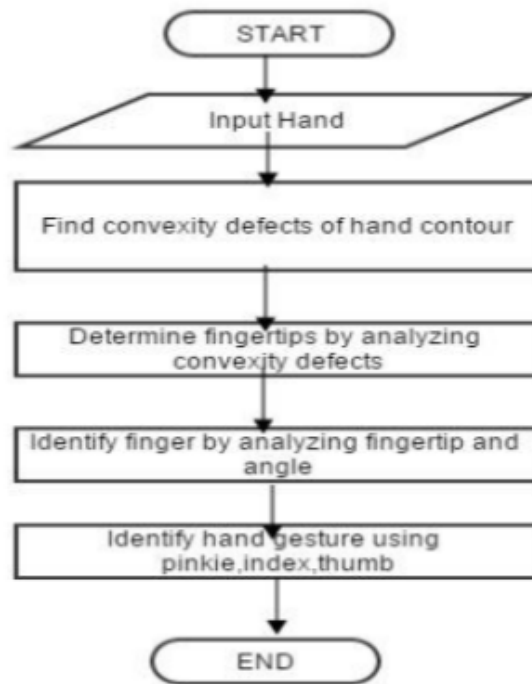


Figure 3.2: Mouse Control Flow Graph



Figure 3.3: Mouse Control Flow Graph

3.3 Voice Control Module

Voice is the basic, common and efficient form of communication method for people to interact with each other. Today speech technologies are commonly available for a limited but interesting range of task. This technologies enable machines to respond correctly and reliably to human voices and provide useful and valuable services. As communicating with computer is faster using voice rather than using keyboard, so people will prefer such system. We add this system to our project in order to make interactions a lot more easier and due to the limitations in using the number of gestures.

3.3.1 Speech Recognition

Speech recognition is the process by which a computer (or other type of machine) identifies spoken words. Basically, it means talking to a computer & having it correctly understand what you are saying. By “understand” we mean, the application to react appropriately or to convert the input speech to another medium of conversation which is further perceivable by another application that can process it properly & provide the user the required result. The days when you had to keep staring at the computer screen and frantically hit the key or click the mouse for the computer to respond to your commands may soon be a things of past. Today we can stretch out and relax and tell your computer to do your bidding. This has been made possible by the ASR (Automatic Speech Recognition) technology.

1. Voice Input

With the help of microphone audio is input to the system, the pc sound card produces the equivalent digital representation of received audio

2. Digitization

The process of converting the analog signal into a digital form is known as digitization [8], it involves the both sampling and quantization processes. Sampling

is converting a continuous signal into discrete signal, while the process of approximating a continuous range of values is known as quantization.

3. Acoustic Model

An acoustic model is created by taking audio recordings of speech, and their text transcriptions, and using software to create statistical representations of the sounds that make up each word. It is used by a speech recognition engine to recognize speech. The software acoustic model breaks the words into the phonemes

4. Language Model

Language modeling is used in many natural language processing applications such as speech recognition tries to capture the properties of a language and to predict the next word in the speech sequence. The software language model compares the phonemes to words in its built in dictionary

5. Speech engine

The job of speech recognition engine is to convert the input audio into text, to accomplish this it uses all sorts of data, software algorithms and statistics. Its first operation is digitization as discussed earlier, that is to convert it into a suitable format for further processing. Once audio signal is in proper format it then searches the best match for it. It does this by considering the words it knows, once the signal is recognized it returns its corresponding text string

The Speech Recognition Software we are using in this project is the Google Web Speech API, Since it has high accuracy and easily available. Some of the voice commands are given below.

Voice Commands	Actions
Shutdown	Shutdown the system
Web browser	Open web browser
Restart	Restart the system
VLC	Open VLC media player

Table 3.3: Voice Commands and their Operations

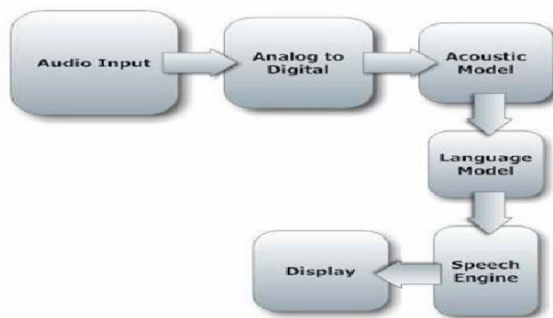


Figure 3.4: Speech Recognition Process

Chapter 4

Results and Discussions

4.1 Skin Segmentation

The figure above shown the output of skin segmentation algorithm. without a hand inside designated green window. As seen below, the skin segmentation algorithm works quite well for the given input. It can easily differentiates between skin and nonskin value. In addition, the computation time is quite fast. This algorithm is processed in realtime without any noticeable delay. This method is based on the hsv values of the skin pixels and we can see from the figures below that on varying lighting conditions the gesture recognition become more difficult since the skin pixel values may change when lights gets into the input and it may affect in identifying the gesture correctly which will end up in opening the application wrongly.

In Figure 4.5,4.6 we can see the that the binary image of the input is not correct since some regions of the skins are not marked. Many methods are adopted and different problems are arised in identfyng gesture. In this project the background remains constant and hence background substraction method can be used to correctly identify the gestures. This method solves the problem upto a limit and it is adapted in this project.The output of the background substraction is given

The 2 methods where tested by ourself with different lighting conditions and different

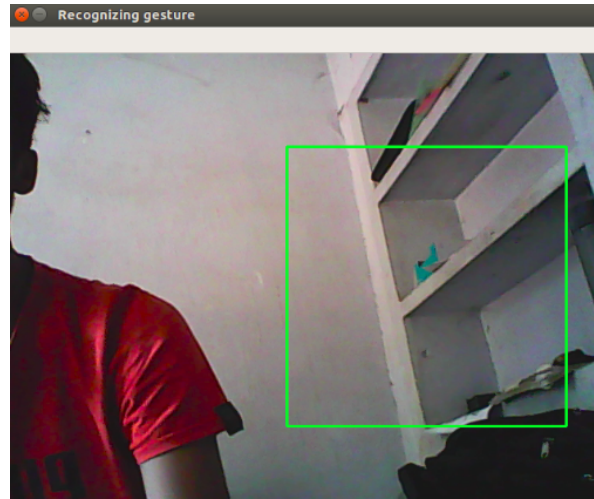


Figure 4.1: Input video frame without a hand inside designated green window.

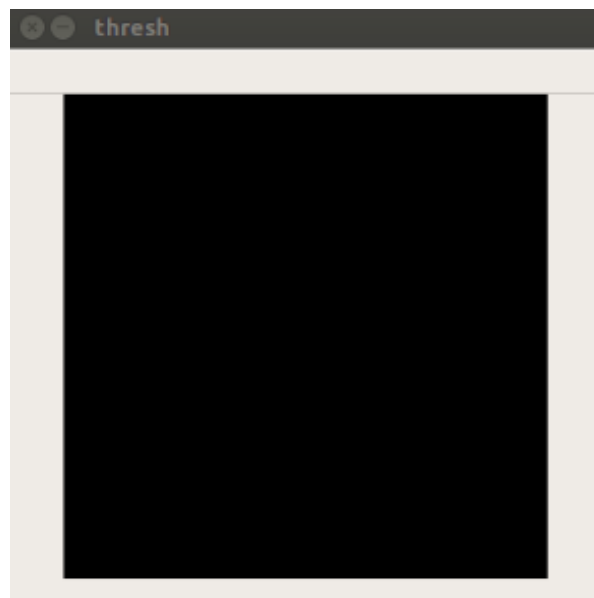


Figure 4.2: Output video frame without a hand inside designated green window.

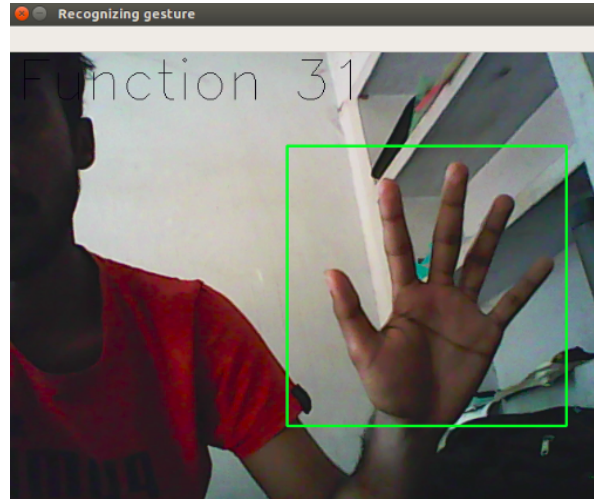


Figure 4.3: Input video frame with a hand inside designated green window



Figure 4.4: Output video frame with a hand inside designated green window

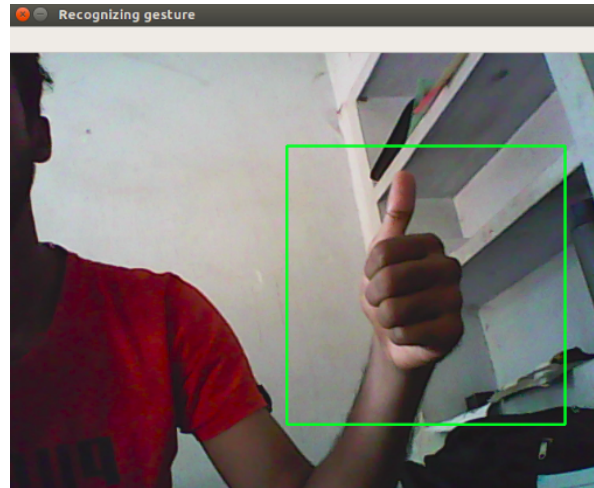


Figure 4.5: Input video frame with a hand inside designated green window



Figure 4.6: Output video frame with a hand inside designated green window

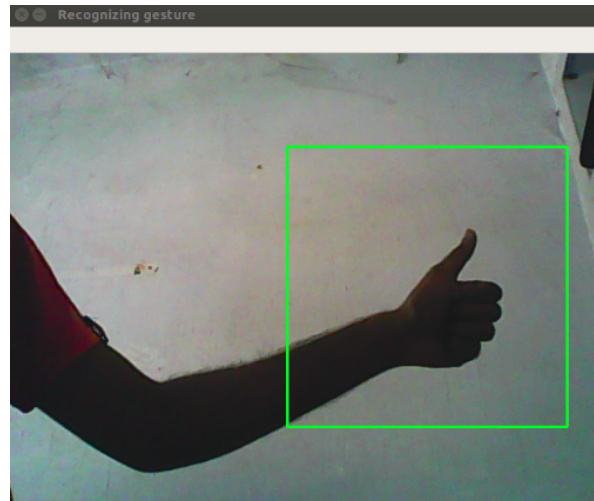


Figure 4.7: Input video frame with a hand inside designated green window using background subtraction method



Figure 4.8: Output video frame with a hand inside designated green window using background subtraction method

Gestures	Background Substraction Method (n=50)	Skin Segmantation using HSV values (n=50)
Gesture A	48	37
Gesture B	50	28
Gesture C	46	41
Gesture D	49	32
Gesture E	49	39
Gesture F	50	38
Gesture G	49	30
Gesture H	48	35
Gesture I	50	37
Gesture J	50	32
Gesture K	49	40
Gesture L	48	40
Gesture M	47	41
Gesture N	50	39
Gesture O	50	38

Table 4.1: Gesture Prediction Result

backgrounds and the result obtained are given below. The 50 input of each gestures were taken and the accuracy with 2 methods are noted

4.2 Mouse Movement Analysis

The mouse movement is based on defect analysis, The convexity defects are found by finding the angle between the two lines. If the angle is greater than 90 it will not be treated as a defect. By considering the angle only for convexity defects which cause the system to mark all the concave part as the defects since we only need the defects between the fingers. We will consider the length of the side also to find the defects. The output of this is given below.

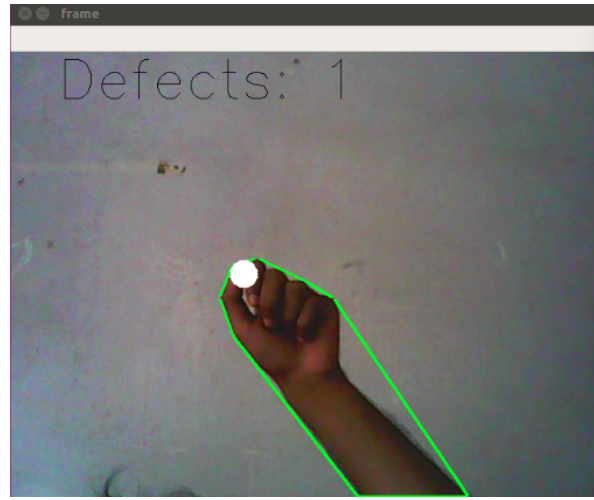


Figure 4.9: Defects of 1 is detected when we use angle measures only

You can see that the part between the folded fingers are taken as a defects. Since it cannot be considered as a defect for our problem.

Comparing the Fig 4.9 with Fig 4.10 we can clearly see that the convexity defect finding mechanism for our problem is worked perfectly when we add the distance measures too.

And another problem faced is the repeated click of mouse when corresponding defects occurred. It is occurred because the webcam continuously processing each frames continuously. It can be avoided by just taking the frame count, That is when the frame of 5 is continuously shows the same defect then only the corresponding action is performed.

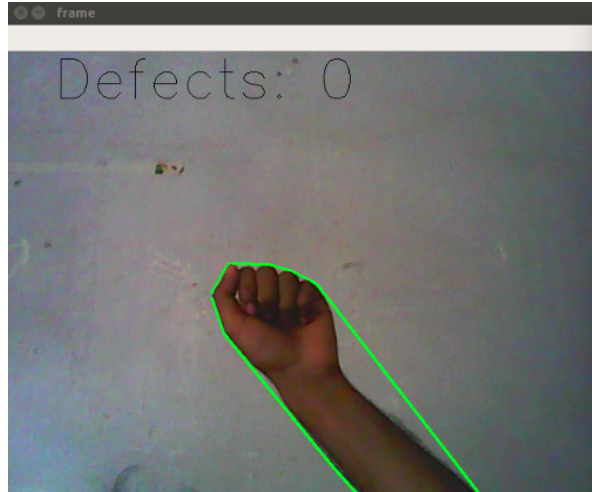


Figure 4.10: Defects of 0 is detected when we use angle measures and distance measures

4.3 Voice Recognition Analysis

The project uses Google Web Speech API for voice recognition, and the result of the Google API is taken from a paper which is described in this analysis. The selected audio files for testing are given in the table below.

File	Original sentences
SX293	please take this dirty table cloth to the cleaners for me
SX223	put the butcher-block table in the garage
SI1894	my father ran him off here six years ago
SI1400	now that this is at odds with our meaning may be shown as follows
SX188	who authorized the unlimited expense account?
SI1628	this is my hen ledger, he informed him in an absorbed way
SI2000	we can get it if we dig, he said patiently
SX216	the small boy put the worm on hook
SX396	the fish began to leap frantically on the surface of the small lake
SI1580	he always seemed to have money in his pocket

Table 4.2: File and the Text on each file

when we give these audio file as input to google API we are get accuracy about 91%.

The detailed analysis of google API is shown below.

Where S sentences, N words, I words were inserted, D words were deleted, and S words

File	S	N	I	S	D	CW	EW	WA	WER
TSX223	1	8	0	0	0	9	0	1.0	0.0
TSX293	1	11	0	1	1	9	2	0.82	0.18
TSi1894	1	9	0	0	0	9	0	1.0	0.0
TSi1400	1	14	0	1	0	13	1	.93	0.07
TSX188	2	6	0	0	0	6	0	1.0	0.0
TSi1628	2	12	0	2	0	10	2	0.83	0.17
TSX314	2	12	0	0	0	12	0	1.0	0.0
DIG001	3	15	0	0	0	15	0	1.0	0.0
TSX216	1	9	0	0	0	9	0	1.0	0.0
TSX209	1	7	0	0	0	7	0	1.0	0.0

Table 4.3: Result of Google API

were substituted. CW correct words, EW error words, WER word error rate.

Chapter 5

Conclusion

Gesture recognition has been an on-going research driven by its wide potential for applications such as sign language recognition, remote control robots and human computer interaction in virtual reality. In this project, we develop a real-time gesture-based Human Computer Interaction system who recognizes gestures only using one camera. The developed system relies on a CNN classifier to learn features and to recognize gestures. We employ a series of steps to process the image and to segment the hand region before feeding it to the CNN classifier in order to improve the performance of the CNN classifier. 3,600 gesture images are used to train the CNN classifier and demonstrate that the CNN classifier combined with our image processing steps can recognize gestures with high accuracy in real time. The usage of the CNN frees us from extracting the gesture features manually and improve the recognition accuracy. Besides, a mouse controlling module is created which is used to control the mouse with our bare hands . We created an another module which identifies user voice and perform the actions according to that. The developed system now only support static gestures. In the future work, we will make a classifiers for dynamic gestures and develop a gesture-based Human Computer Interaction or Human Robotic Interaction system with the support of complex motion recognition.

Bibliography

- [1] R. M. Gurav and P. K. Kadbe, “Real time finger tracking and contour detection for gesture recognition using opencv,” in *2015 International Conference on Industrial Instrumentation and Control (ICIC)*, pp. 974–977, IEEE, 2015.
- [2] M. Shanmukhi, K. L. Durga, M. Mounika, and K. Keerthana, “Convolutional neural network for supervised image classification,” *International Journal of Pure and Applied Mathematics*, vol. 119, no. 14, pp. 77–83, 2018.
- [3] M. Beena, M. A. Namboodiri, and P. Dean, “Automatic sign language finger spelling using convolution neural network: analysis,” *International Journal of Pure and Applied Mathematics*, vol. 117, no. 20, pp. 9–15, 2017.
- [4] V. Kėpuska and G. Bohouta, “Comparing speech recognition systems (microsoft api, google api and cmu sphinx),” *Int. J. Eng. Res. Appl*, vol. 7, no. 03, pp. 20–24, 2017.