JAWAHARLAL NEHRU NATIONAL COLLEGE OF ENGINEERING

SHIMOGA - 577204

Department of Telecommunication Engineering

CERTIFICATE

This is to certify that the project work entitled

"An Image Processing Technique For The Translation Of Indian Sign Language Finger-Spelling To Audio And Text"

Is a Bonafide work carried out jointly by

DHENUKA M	**4JN11TE009**
SHREEKANTHA A N	**4JN10TE038**
SPOORTHI M N	**4JN10TE047**
SUJAY SHARMA	**4JN10TE051**

These Students of 8th semester B.E., Telecommunication Engineering under our supervision and guidance towards the partial fulfillment of the requirements of the award of the degree of Bachelor of Engineering in Telecommunication Engineering as per the University regulations during the year 2014.

……………………	..………………………	…………………………..
Signature of the Guide	Signature of the HOD	Signature of principal
Sandeep Kumar E	**Prof. Amarappa S.**	**Dr.Srinivasa Rao Kunte**
Asst. professor	**M.Tech., (Ph.D)**	**M. Tech.,Ph.D**

External Viva

Name of the Student:

USN:

Name of Examiners: **Signature of Examiners with date:**

1. ………………………….. **..…………………………...**

2. ………………………….. **...…………………………..**

ABSTRACT

A sign language is a language which uses manual communication and body language to convey meaning, as opposed to acoustically conveyed sound patterns. This can involve simultaneously combining hand shapes, orientation and movement of the hands, arms or body, and facial expressions to fluidly express a speaker's thoughts. Sign language is used by hearing impaired people to communicate. This sign language is difficult and not understood by many people.

Sign Language Recognition is helpful in communication between hearing impaired people and others who are not familiar with sign language. Sign Language Recognition has emerged as one of the important area of research in Computer Vision.

The main aim of this project is to develop an automating Translation of Indian sign language, static single handed character signs. The system is designed to translate signs, i.e. each input image is obtained by video feed of Indian Sign Language (ISL) character sign. The proposed methodology comprises of four stages: Acquisition, Preprocessing, Feature Extraction and Classification.

The approach of using bare hands allows the user to interact with the system in natural way. We have considered 24 different alphabets in the video sequences. This project shall provide the design approach on both software and hardware.

ACKNOWLEDGEMENT

The euphoria of successful completion of the project **“An Image Processing Technique For The Translation Of Indian Sign Language Finger-Spelling to Audio and Text”** would be incomplete without mentioning the names of those people who have helped us to achieve this milestone. We take this opportunity to express our gratitude to all those who have helped us in all ways towards the successful completion of this project.

We express our special thanks to **Dr. Srinivasa Rao Kunte**, Principal of JNNCE, Shimoga for providing us an opportunity to undertake this project.

We sincerely thank **Prof. Amarappa S**, Head of the Department, for providing us with adequate facilities, ways and means by which we are able to complete this project.

We express our profound gratitude to our project guide **Mr. Sandeep Kumar E,** Assistant Professor, Department of Telecommunication Engineering, for his inspirational guidance during the course of work.

We extend our heartfelt gratitude to the Faculty of the Department of Telecommunication Engineering for their constructive support to present this project.

Finally, we thank our families and friends for their tremendous support throughout our educational career till now. We also like to thank them for their faith, they have in us.

- Project Associates

Dhenuka M

Shreekantha A N

Spoorthi M N

Sujay Sharma

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

PREAMBLE

1.1 Introduction

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image. Usually **Image Processing** system includes treating images as two dimensional signals while applying already set signal processing methods to them.

The two types of methods used for Image Processing are Analog and Digital Image Processing. Analog or visual techniques of image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. The image processing is not just confined to area that has to be studied but on knowledge of analyst. Association is another important tool in image processing through visual techniques. So analysts apply a combination of personal knowledge and collateral data to image processing.

Digital image processing is the use of computer algorithms to perform image processing on digital images. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be modeled in the form of multidimensional systems.

Digital Image Processing is among rapidly growing technologies today, with its applications in various aspects. It forms the core research area in many engineering disciplines.

1.2 Problem Statement

A Sign Language is a language which is used to communicate between people by using sign patterns to express the meaning. It is used by **hearing impaired people**. Thus, it has attracted many researchers to this field. Many researchers have been working on different sign languages like American Sign Language, British Sign Language, Taiwanese Sign Language, etc. But a very few progress has been made on Indian Sign Language.

The hearing impaired people are being neglected from the society because **the normal people neither try to learn Sign Language nor try to interact with the hearing impaired people**. This has become the greatest hurdle for them which has made them remain uneducated and isolated.

Our project aims to bridge the gap of communication between the hearing impaired people and the others. We aim to designing an image processing technique for the translation of this sign language into audio and text. Thus, the **translation of sign language** is not only important from engineering point of view but also for the better impact on society.

1.3 Methodology

The algorithm for translation of sign language is implemented by using **Image Processing Toolbox** in **Matlab 8.2**. **Raspberry Pi 2** is used for the **hardware implementation** of the same algorithm. Translation of sign language includes Image Acquisition, Pre-processing, Feature Extraction and Classification.

Image Acquisition: The images of signs are taken from the 5MP Pi camera. All the images are taken from front view with plain light background. Video feed is acquired from the camera through which images are sampled at a maximum rate of frames 60 frames per second. We have used sampling rate of 30 frames per second. Images thus obtained are further preprocessed.

Pre-processing: Images obtained are pre-processed to extract only hand showing the sign. Skin Filtering technique is used for this purpose.

Feature Extraction: Histogram of Gradients (HOG) Features are extracted for the preprocessed images. This HOG feature is unique for each sign and it is a global feature for all each sign.

Classification: Support Vector Machine (SVM) is one of the supervised learning techniques used for classification analysis. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier.

1.4 Scope of the project

This project can be seen as an application of gesture recognition using image processing with 24 different signs. It can be used to recognize and classify the sign language.

This project aims to bridge the gap of communication existing between the hearing impaired people and others. Hence, the project is not only for the purpose for executing the engineering knowledge but also for better impact on huge mass of people.

1.5 Limitations

This project has the following limitations.

- Translation is not possible for dynamic signs ‘J’ and ‘Z’.
- Results are dependent on the lighting condition.
- Translation of signs has a minor delay.

1.6 Organization of the report

This report is organized into 5 chapters. The first chapter gives a brief introduction about the project. The second chapter brings out a clear idea of theoretical aspects and all concepts which inspires to take up the project. The third chapter is the heart of the project which gives the details about the design and implementation of the project. The fourth chapter deals with information about the results. The fifth chapter concludes with the future scope of the project.

1.7 Summary

This chapter provided the basic information about the need of translation of sign language. It presented precise information of the methodology used, scope and limitations of the project. The structure of sign language discussed in the next chapter.

CHAPTER 2

THEORETICAL BACKGROUND

This chapter gives an idea about the Sign Language, Image Processing Technique, and Support Vector Machine (SVM). The block diagram is explained necessarily.

2.1 The Sign Language

A sign language (also signed language or simply signing) is a language which uses manual communication and body language to convey meaning, as opposed to acoustically conveyed sound patterns. This can involve simultaneously combining hand shapes, orientation and movement of the hands, arms or body, and facial expressions to fluidly express a speaker's thoughts. They share many similarities with spoken languages, which is why linguists consider both to be natural languages, but there are also some significant differences between signed and spoken languages.

Figure 2.1: People communicating in Sign Language.

[*Courtesy: Jasign Expressions, American Sign Language Training Institute.]*

This section comprises of introduction to sign language, history of sign language, relationship with spoken language, list of sign languages and alphabets of ISL.

Wherever communities of hearing impaired people exist, sign languages have been developed. Signing is not only used by the hearing impaired, it is also used by people who can hear, but cannot physically speak. While they use space for grammar in a way that spoken languages do not, sign languages show the same linguistic properties and use the same language faculty as do spoken languages.[1] Hundreds of sign languages are in use around the world and are at the cores of local hearing impaired cultures. Some sign languages have obtained some form of legal recognition, while others have no status at all.

A common misconception is that all sign languages are the same worldwide Aside from the pidgin International Sign, each country generally has its own, native sign language, and some have more than one, though sign languages may share similarities to each other, whether in the same country or another one.

It is not clear about how many sign languages there are. The 2013 edition of Ethnologue lists 137 sign languages. [2]

2.1.1 History of Sign Language

The recorded history of sign language in Western societies starts in the 17th century, as a visual language or method of communication. Sign language is composed of a system of conventional gestures, mimic, hand signs and finger spelling, plus the use of hand positions to represent the letters of the alphabet. Signs can also represent complete ideas or phrases, not only individual words. Most sign languages are natural languages, different in construction from oral languages used in proximity to them, and are employed mainly by hearing impaired people in order to communicate.

One of the earliest written records of a sign language is from the fifth century BC, in Plato's Cratylus, where Socrates says: "If we hadn't a voice or a tongue, and wanted to express things to one another, wouldn't we try to make signs by moving our hands, head, and the rest of our body, just as dumb people do at present?"[3]

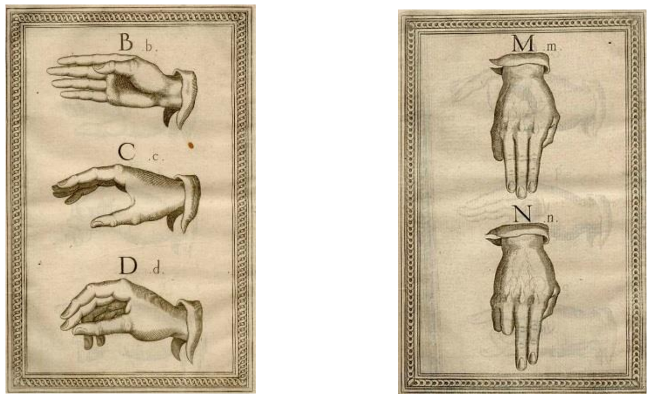


Figure 2.2: Demonstration of BCDMN 'Reduction of letters and art to teach the dumb to speak' [*Courtesy: National Library of Spain, Juan Pablo Bonet*]

Until the 19th century, most of what known about historical sign languages is limited to the manual alphabets (fingerspelling systems) that were invented to facilitate transfer of words from a spoken to a signed language, rather than documentation of the rest of the language. However, many different signed languages have developed independently throughout the world, and no true first language can be identified. Both signed systems and manual alphabets were found worldwide, and, though most recorded instances of signed languages seem to occur in Europe in the 17th century, it is possible that popular European ideals have overshadowed much of the attention earlier signed systems may have otherwise received.

It was commonly accepted, for instance, that "the hearing impaired" could not be educated; when John of Beverly, Archbishop of York, taught a hearing impaired person to speak in 685 AD, it was deemed a miracle, and he was later canonized [2]. Generally, philosophies linking (spoken) language and intelligence persisted well into the Enlightenment. Such hegemonic ideas may have prevented the recognition of histories of certain groups for whom signed languages were integral.

Deaf-community sign languages arise where hearing impaired people come together to form their own communities. These include school sign which develop in the student bodies of deaf schools which do not use sign as a language of instruction. At first, Deaf-community sign languages are not generally known by the hearing population, in many cases not even by close family members. However, they may grow, in some cases becoming a language of instruction and receiving official recognition, as in the case of ASL, BSL and ISL. [3]

2.1.2 Relationship with Spoken Language

Sign languages generally do not have any linguistic relation to the spoken languages. The correlation between sign and spoken languages is complex and varies depending on the country more than the spoken language.

As a sign language develops, it sometimes borrows elements from spoken languages. Sign languages vary in how and how much they borrow from spoken languages. In many sign languages, a manual alphabet (fingerspelling) may be used in signed communication to borrow a word from a spoken language, by spelling out the letters. This is most commonly-used for proper names of people and places; it is also used in some languages for concepts for which no sign is available at that moment, particularly if the people involved are to some extent bilingual in the spoken language. **Fingerspelling** can sometimes be a source of new signs, such as initialized signs, in which the handshape represents the first letter of a spoken word with the same meaning.

On the whole, though, sign languages are **independent** of spoken languages and follow their **own paths of development**. British Sign Language and American Sign Language (ASL) are quite different and **mutually unintelligible**. The grammars of sign languages do not usually resemble that of spoken languages used in the same geographical area.

2.1.3 List of Sign Languages

There are three hundred sign languages in use around the world. The number is not known with any confidence. In some countries, such as Sri Lanka and Tanzania, each school for the hearing impaired may have a separate language, known only to its students. Some countries may share sign languages which sometimes are under different names (Croatian and Serbian, Indian and Pakistani). Deaf sign languages also arise outside of educational institutions, especially in village communities with high levels of congenital deafness, but there are significant sign languages developed for the hearing as well, such as the speech-taboo languages used in aboriginal Australia. Scholars are doing field surveys to identify the world's sign languages. [4]

The following list is grouped into three sections: [3]

- **Deaf sign languages**, which are the preferred languages of Deaf communities around the world; these include village sign languages, shared with the hearing community, and Deaf-community sign languages
- **Auxiliary sign languages**, which are not native languages, but are signed language systems of varying complexity, used in addition to spoken languages. Simple gestures are not included, as they do not constitute language.
- **Signed modes of spoken languages**, also known as manually coded languages, which are bridges between signed and spoken languages

The sign languages are sorted regionally besides having same origin. Some of them can be listed in table 2.1. [4]

Language	Origin
a) American Sign Language (ASL)	French
b) Algerian Sign Language	French
c) British Sign Language (BSL)	British
d) Brazilian Sign Language	Brazilian
e) Chinese Sign Language	Chinese
f) Indian Sign Language (ISL)	ASL
g) Kenyan Sign Language	Local Origin
h) Libyan Sign Language	Arabian
i) Northern Ireland Sign Language	BSL
j) Portuguese Sign Language	Swedish
k) Yemeni Sign Language	Algerian Sign Language
l) Zimbabwean sign languages	Not known

Table 2.1: List of sign languages with their origin.

2.1.4 Indian Sign Language and Its Alphabets

Indian Sign Language (ISL) has its origin from American Sign Language (ASL). It was found that count of hearing impaired people in India, is more compared to other countries. Not all of them use ISL but, more than one million deaf adults and around half million deaf children use ISL as a mode of communication. Hearing impaired people, who live in villages usually, do not have access to sign language. However, in all large towns and cities across the Indian subcontinent, these people use sign language which is not standard sign language.

ISL is broadly categorized into 2 categories. They are

- Manual (hand shape, hand location, orientation and movements)
- Non-manual (facial expression, eye gaze and head/body posture).

In ISL, there are **one handed** and **two handed** signs which can be **static** and **dynamic**. Figure 2.3 shows ISL manual alphabet set. [5] We have considered **One handed Manual Static** alphabets.

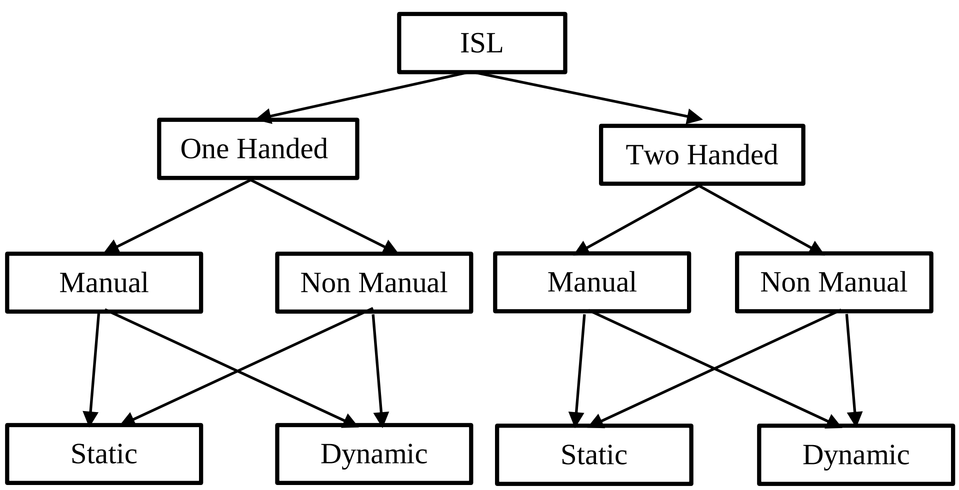


Figure 2.3: A Hierarchical Classification of ISL.

Figure 2.4: Manual One Handed ISL.

2.2 Block Diagram and Working Principle

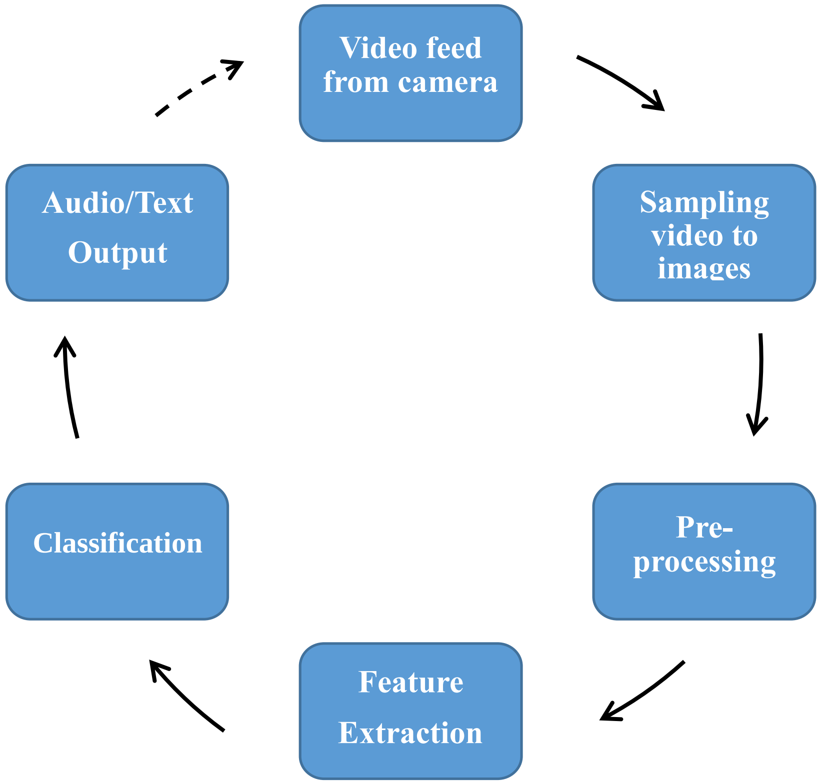


Figure 2.5: Block diagram for the translation of ISL.

The proposed methodology has four major parts such as Image Acquisition from live video, Pre-processing of the acquired image, Feature Extraction and Classification. The output of the classification can be seen on LCD Display and heard through Speakers. The block diagram of proposed methodology is as shown in figure 2.5.

Images needed for recognition of sign is obtained by sampling the live video acquired by the camera. These images are pre-processed to extract only the hand part from whole of the image. Further, HOG Features are extracted in feature extraction stage to classify each sign. Using these features extracted, classification of each sign is done by making use of SVM classifier. The text and audio output is triggered when the signs are classified.

The software implementation of this project is done using MATLAB 8.2 release name R2013b. And hardware implementation is taken up on Raspberry Pi 2 programmed in Python 2.4 in OpenCV. Each of the concepts involved in this project are explained in detailed in the following sections.

2.3 Concepts of Image Processing

Image processing is any form of signal processing for which the input is an **image**, such as a photograph or video frame. The output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it.

Image processing usually refers to **digital image processing**. In this section, description of image processing technique used in the project is explained.

The steps involved in image processing part of our project are

- Image Acquisition
- Pre-processing
- Feature Extraction

The techniques employed in our project for each of these stages to carry out are explained in the following sections.

2.3.1 Image Acquisition

Image acquisition in image processing can be broadly defined as the action of retrieving an image from some source. Performing image acquisition in image processing is always the first step in the workflow sequence because without an image no processing is possible. The image that is acquired is completely is the result of whatever hardware was used to generate it, which can be very important in some fields to have a consistent baseline from which to work. One of the ultimate goals of this process is to have a source of input that operates within such controlled and measured guidelines that the same image can, if necessary, be nearly perfectly reproduced under the same conditions so anomalous factors are easier to locate and eliminate.

Depending on the field of work, a major factor involved in image acquisition in image processing sometimes is the initial setup and long-term maintenance of the hardware used to capture the images. The actual hardware device can be anything from a desktop scanner to a massive optical telescope. If the hardware is not properly configured and aligned, then visual artifacts can be produced that can complicate the image processing. Improperly setup hardware also may provide images that are of such low quality that they cannot be salvaged even with extensive processing. All of these elements are vital to certain areas, such as comparative image processing, which looks for specific differences between image sets.

One of the forms of image acquisition in image processing is known as **real-time image acquisition**. This usually involves retrieving images from a source that is automatically capturing images. Real-time image acquisition creates a stream of files that can be automatically processed, queued for later work, or stitched into a single media format.

The images can be acquired in many color spaces like RGB, YCbCr, HSV, HIS, Grey Scale, CMY. The images can be directly acquired in many of these colorscpaces or can be converted during processing in order to get desired processed image.

We have used real-time image acquisition by sampling images from live video feed. The images captured are in RGB colorspace.

2.3.2 Pre-processing

Pre-processing is a common word for operation on image in lowest level involving correction of distortion, degradation, noise and changing the colorspace introduced during the imaging process.

The main of pre-processing is an improvement of image data that has unwanted distortion or some important feature that is needed for further processing. But pre-processing has a drawback that is the enhancement techniques can emphasize image artefacts, or even lead to a loss of information if not correctly used. So care should be taken while for taking up each pre-processing technique such that no much information is lost.

Some of the techniques used in pre-processing are image resampling, image resampling, greyscale contrast enhancement, noise removal, mathematical operations and manual correction.

The techniques used for pre-processing in this project are

- RGB to HSV conversion for skin filtering
- Grey scale conversion for HOG feature extraction

RGB to HSV Conversion

The Hue Saturation Value model was created by A. R. Smith in 1978. It is based on such intuitive color characteristics as tint, shade and tone (or family, purety and intensity). The coordinate system is cylindrical, and the colors are defined inside a hexcone. The hue value H runs from 0 to 360º. The saturation S is the degree of strength or purity and is from 0 to 1. Purity is how much white is added to the color, so S=1 makes the purest color (no white). Brightness V also ranges from 0 to 1, where 0 is the black. Figure 7 shows HSV colorspace visualization.

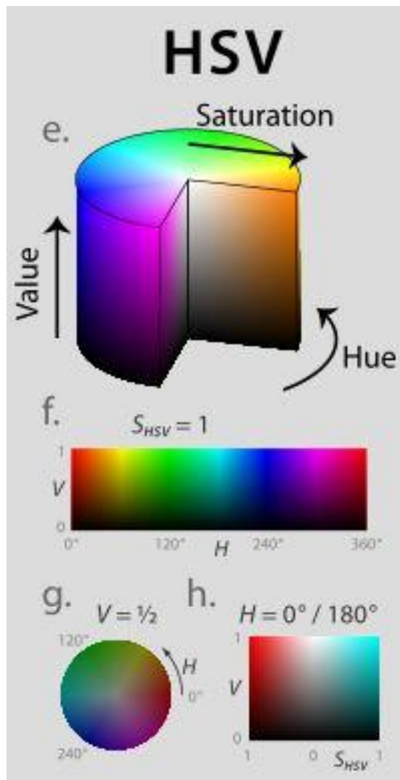


Figure 2.6: HSV cut-away 3D models with two-dimensional plots showing two of a model's three parameters at once.

HSV is a cylindrical geometries (fig. 2.6), with hue, it's angular dimension, starting at the red primary at 0°, passing through the green primary at 120° and the blue primary at 240°, and then wrapping back to red at 360°. In each geometry, the central vertical axis comprises the neutral, gray colors, ranging from black at lightness 0 or value 0, the bottom, to white at lightness 1 or value 1, the top as shown in figure 8. The additive primary and secondary colors – **red, yellow,**

green, cyan, blue, and magenta – and linear mixtures between adjacent pairs of them, sometimes called pure colors, are arranged around the outside edge of the cylinder with saturation 1; in HSV these have value 1. In HSV, mixing these pure colors with white – producing so-called tints – reduces saturation, while mixing them with black – producing shades – leaves saturation unchanged.

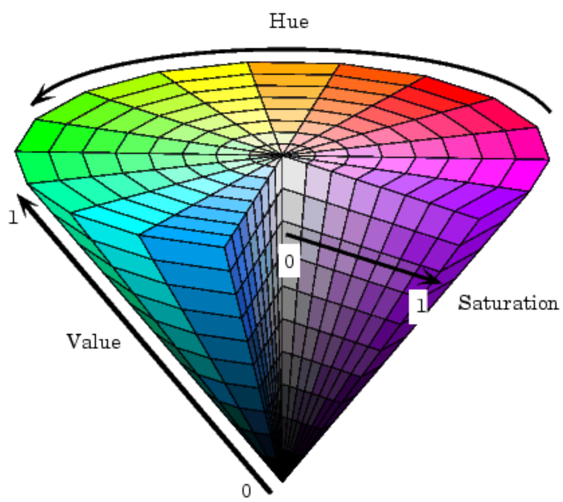


Figure 2.7: HSV Hexcone.

We have used RGB to HSV conversion in order to obtain skin filter from whole of the captured image. HSV is commonly found in all computer vision applications. The color displayed on one monitor for a given HSV value is unlikely to exactly match the color seen on another monitor unless the two are precisely adjusted to **absolute color spaces**. [8]

The algorithm for RGB to HSV conversion is as follows. It has no transformation matrix.

R' = R/255 ##The R,G,B values are divided by 255 to

G' = G/255 ## change the range from 0-255 to 0-1

B' = B/255

Cmax = max(R', G', B'); Cmin = min(R', G', B'); Δ = Cmax - Cmin

Hue calculation:

$$H = \begin{cases} 6o^0 \text{ x } ((G'-B'/\Delta) \bmod 6)\ , & Cmax = R' \\ 6o^0 \text{ x } ((B'-R'/\Delta) \bmod 6)\ , & Cmax = G' \\ 6o^0 \text{ x } ((R'-G'/\Delta) \bmod 6)\ , & Cmax = B' \end{cases}$$

Saturation calculation:

$$S = \begin{cases} 0 & , C_{max} = 0 \\ \Delta / C_{max} & , C_{max} \neq 0 \end{cases}$$

Value calculation:

$$V = C_{max}$$

An illustration of RGB to HSV conversion in H, S and V planes can be shown in figure 2.8.

Figure 2.8: An illustration of RGB to HSV conversion.

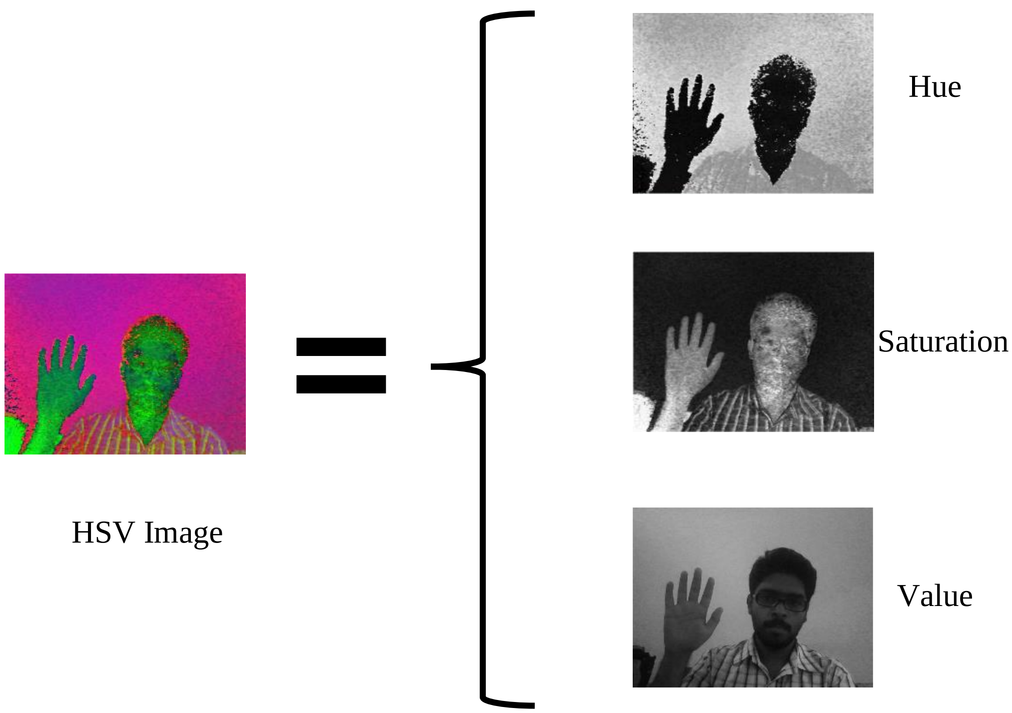


Figure 2.9: H, S and V layers of HSV image.

RGB to Grey Scale Conversion

A grayscale or greyscale digital image is an image in which carries only **intensity** information. Images of this sort, also known as black-and-white, are composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest. Grayscale images are distinct from one-bit bi-tonal black-and-white images, black, and white (also called bilevel or binary images). Grayscale images have many shades of gray in between.

The intensity of a pixel is expressed within a given range between a minimum and a maximum. This range is represented in a way as a range from 0 (total absence, black) and 1 (total presence, white), with any fractional values in between. Figure 2.10 shows RGB to Greyscale conversion below.

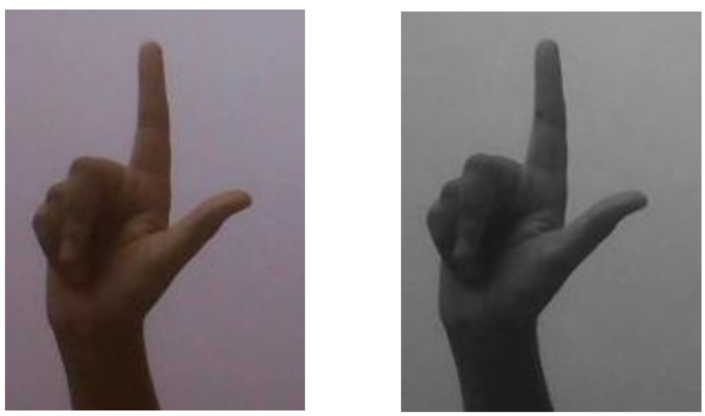


Figure 2.10: RGB to Greyscale converted image of alphabet L.

2.3.3 Feature Extraction

Feature extraction involves reducing the amount of resources required to describe a large set of data. Feature extraction is a type of dimensionality reduction that efficiently represents interesting parts of an image as a compact feature vector. This approach is useful when image sizes are large and a reduced feature representation is required to quickly complete tasks such as image matching and retrieval. It is the most important step in image processing.

Many data analysis software packages provide for feature extraction and dimension reduction. Common numerical programming environments such as MATLAB, SciLab, NumPy and the R language provide some of the simpler feature extraction techniques (e.g. principal component analysis) via built-in commands.

Common feature extraction techniques include Histogram of Oriented Gradients (HOG), Speeded Up Robust Features (SURF), Local Binary Patterns (LBP), Haar wavelets, and color histograms.

We have considered Histogram of Oriented Gradients (HOG) in our project to obtain features of each signs.

Histogram of Oriented Gradients

The histogram of oriented gradients (HOG) is a feature descriptor used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of an image. This method is similar to that of edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts, but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy.

Navneet Dalal and Bill Triggs, researchers for the French National Institute for Research in Computer Science and Control (INRIA), first described HOG descriptors at the 2005 Conference on Computer Vision and Pattern Recognition (CVPR).[9]

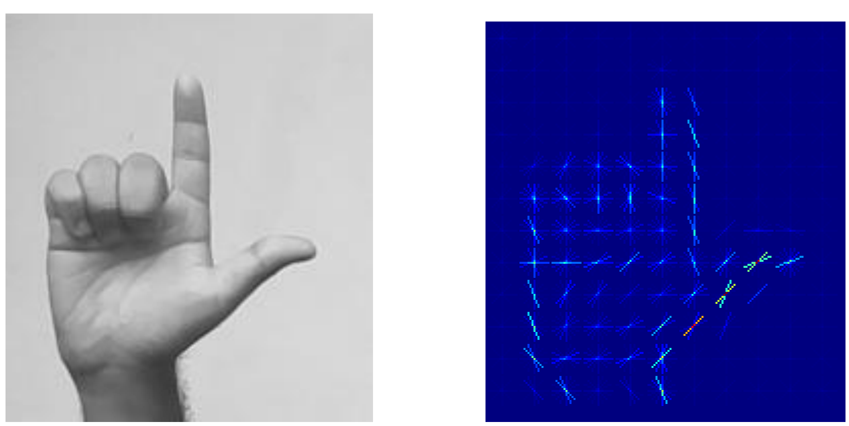


Figure 2.11: HOG features extracted for alphabet L.

The essential thought behind the histogram of oriented gradients descriptor is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. The image is divided into small connected regions called cells, and for the pixels within each cell, a histogram of gradient directions is compiled. The descriptor is then the concatenation of these histograms. For improved accuracy, the local histograms can be contrast-normalized by calculating a measure of the intensity across a larger region of the image, called a block, and then using this value to normalize all cells within the block. This normalization results in better invariance to changes in illumination and shadowing.

The HOG descriptor has a few key advantages over other descriptors. Since it operates on local cells, it is invariant to geometric and photometric transformations, except for object orientation. Such changes would only appear in larger spatial regions.

The implementation of the HOG descriptor algorithm is as follows: [9]

- Divide the image into small connected regions called cells, and for each cell compute a histogram of gradient directions or edge orientations for the pixels within the cell.
- Discretize each cell into angular bins according to the gradient orientation.
- Each cell's pixel contributes weighted gradient to its corresponding angular bin.
- Groups of adjacent cells are considered as spatial regions called blocks. The grouping of cells into a block is the basis for grouping and normalization of histograms.
- Normalized group of histograms represents the block histogram. The set of these block histograms represents the descriptor.

The implementation of HOG descriptor algorithm can be better understood by figure 2.12.

HOG feature extraction is efficient technique for object detection. Since each sign in sign language is distinct, HOG descriptor for each sign obtained helps is recognition of each sign. Thus, HOG feature is being used in this project for better classification of signs.

The final step in recognition of signs using HOG descriptors is to feed the descriptors into some recognition system based on supervised learning. The **support vector machine (SVM) classifier** is a binary classifier which looks for an optimal hyperplane as a decision function. Once trained on images containing some particular object, the SVM classifier can make decisions regarding the presence of an object, such as a human, in additional test images. The next section discusses about SVM classifier.

\

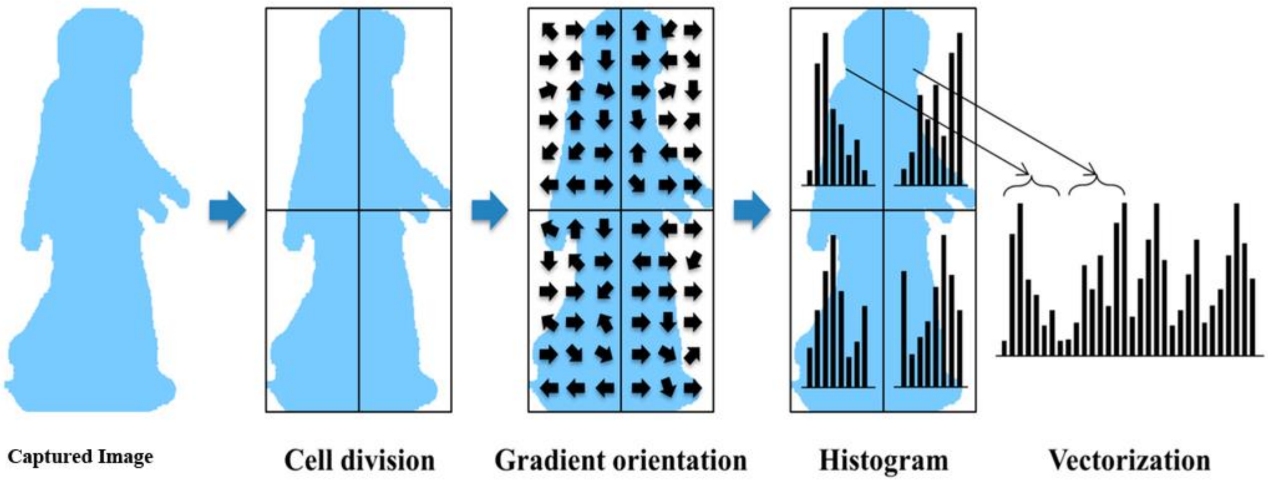


Figure 2.12: HOG descriptor algorithm.

2.3.4 Classification

Classification between the objects is an easy task for humans but it has proved to be a complex problem for machines. Classification step categorizes detected image objects into predefined classes by using suitable method that compares the image patterns with the target patterns. The objective of image classification is to identify a unique feature occurring in an image.

Classification analyzes the numerical properties of various image features and organizes data into categories. Classification algorithms typically employ two phases of processing: **training and testing**. In the initial training phase, characteristic properties of typical image features are isolated and, based on these, a unique description of each classification category, i.e. training class, is created. In the subsequent testing phase, these feature-space partitions are used to classify image features.

Many parameter decides the classification mechanism which can possibly help for classification. [10] Some of them are as follows:

- The Basis Of Characteristic Used
- The Basis Of Training Sample Used
- The Basis Of Assumption Of Parameter on Data
- The Basis Of Pixel Information Used

The most widely used classification techniques for image processing are, [10]

- Artificial Neural Network
- Decision tree
- Support Vector Machine
- Fuzzy Measure

We have used Support Vector Machine (SVM) for classification of signs in this project. The detailed description about SVM is in next section.

Support Vector Machine

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection. SVM can be used when the data has exactly two classes. An SVM classifies data by finding the **best hyperplane** that separates all data points of one class from those of the other class. The best hyperplane for an SVM means the one with the largest margin between the two classes. Margin means the maximal width of the slab parallel to the hyperplane that has no interior data points.

The support vectors are the data points that are closest to the separating hyperplane; these points are on the boundary of the slab. The figure 2.13 illustrates these definitions, with red dot indicating data points of type -1, and yellow squares indicating data points of type 1.

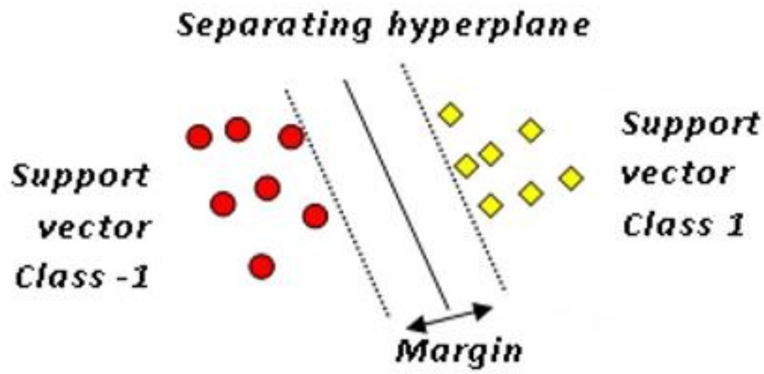


Figure 2.13: Illustration of SVM.

SVMs belong to a family of generalized linear classifiers and can be interpreted as an extension of the perceptron. A special property is that they **simultaneously minimize the empirical classification error and maximize the geometric margin**; hence they are also known as **maximum margin classifiers**.

The operation of the SVM algorithm is based on finding the hyperplane that gives the largest minimum distance to the training examples. Twice, this distance receives the important name of margin within SVM's theory. Therefore, the optimal separating hyperplane maximizes the margin of the training data.

The **advantages** of support vector machines are:

- Effective in high dimensional spaces.
- Still effective in cases where number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different **Kernel Functions** can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The **disadvantages** of support vector machines include:

- If the number of features is much greater than the number of samples, the method is likely to give poor performances.

In this project, SVM is used by considering 1000 samples for each sign for training SVM. All the training samples considered are taken with plane background. Further details of SVM training and testing is explained in next chapter.

The most important aspect of this project is to know about the hardware which is used. We have made use of Raspberry Pi 2 for the hardware implementation of the project. Next section is comprises theoretical explanation of Raspberry Pi 2.

Classifier Comparison

A comparison of a several classifiers in scikit-learn on synthetic datasets. The point of this example is to illustrate the nature of decision boundaries of different classifiers. In high-dimensional spaces, data can more easily be separated linearly and the simplicity of classifiers such as naive Bayes and linear SVMs might lead to better generalization than is achieved by other classifiers.

The plots show training points in solid colors and testing points semi-transparent. The lower right shows the classification accuracy on the test set.

The hardest part of solving a machine learning problem can be finding the right estimator for the job. Different estimators are better suited for different types of data and different problems.

The flowchart in figure 2.15 shows users a bit of a rough guide on how to approach problems with regard to which estimators to try on our data.

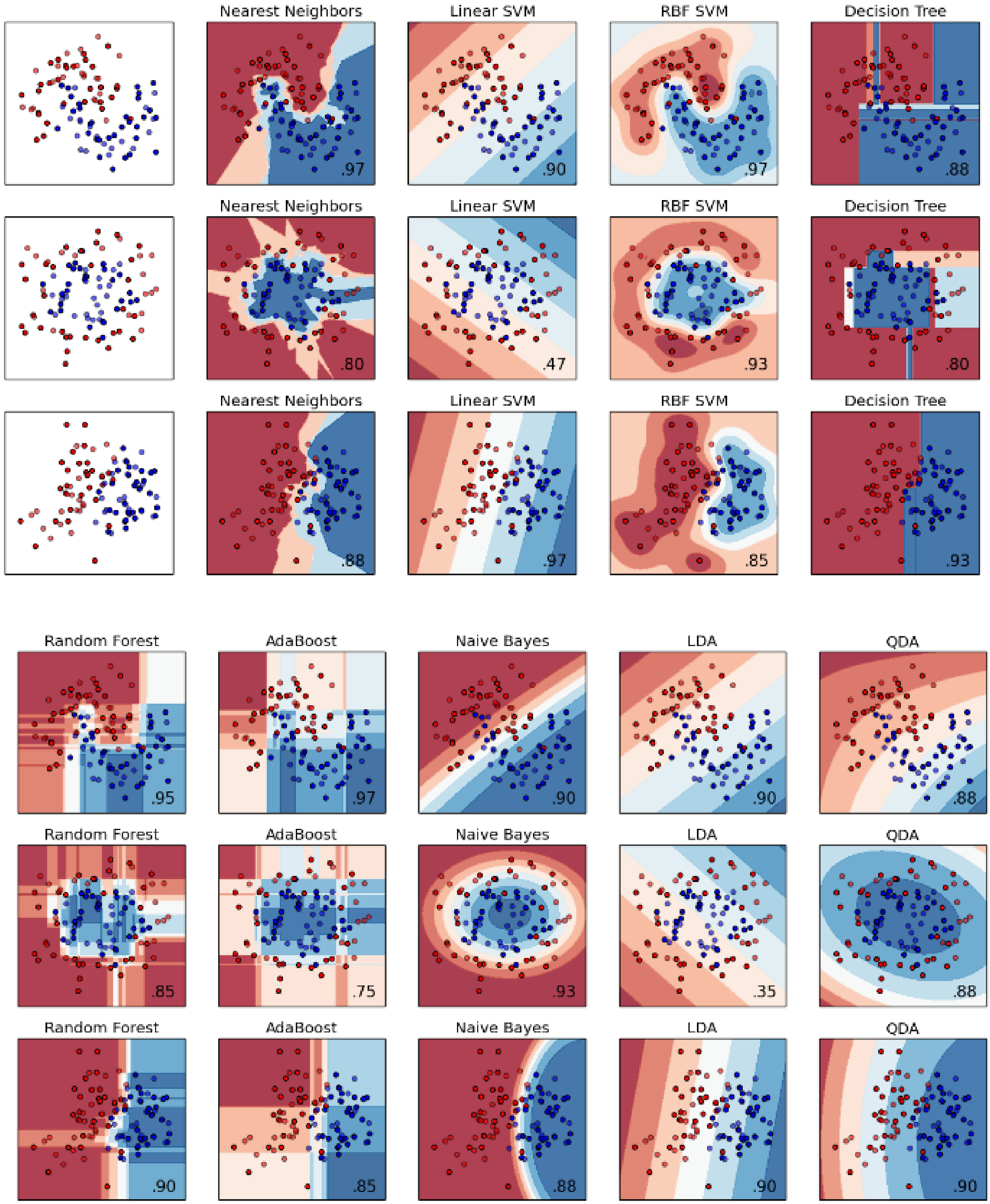


Figure 2.14: Comparison of classifiers in scikit-learn.

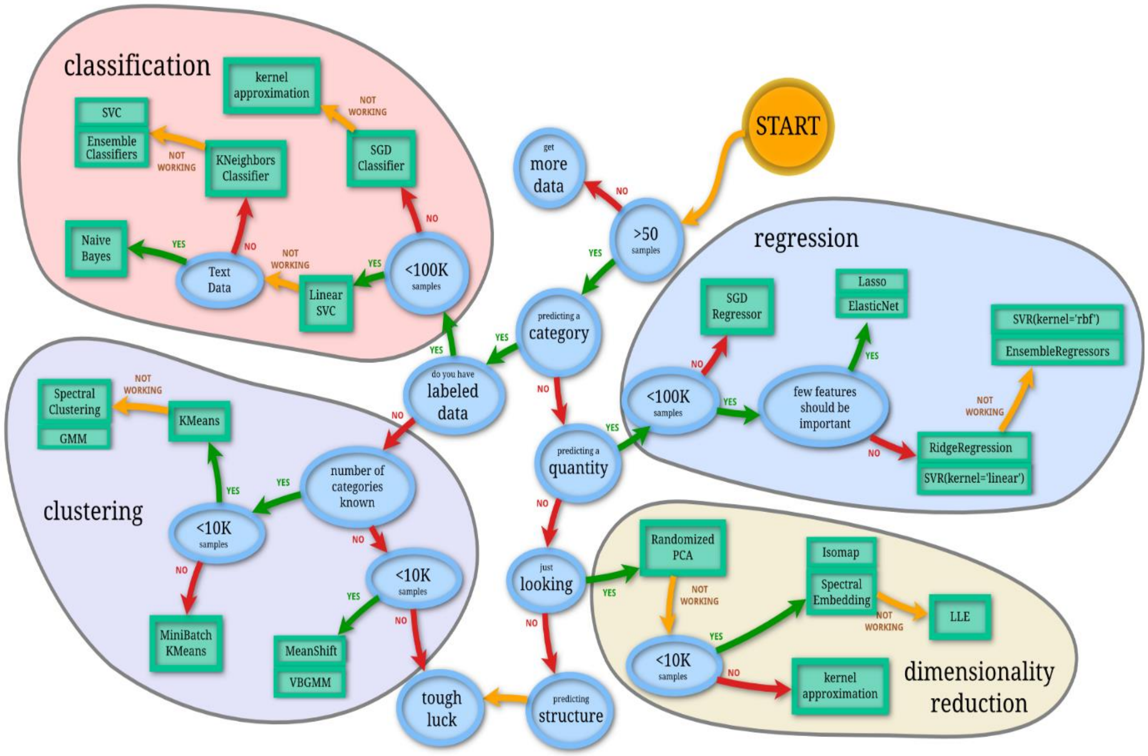


Figure 2.15: Flowchart to approach problems to estimate data.

2.5 Summary

A detailed discussion of the concepts of image processing techniques used in the project were discussed in this chapter. The following chapter will focus on the design aspects and its implementation.

CHAPTER 3

DESIGN AND IMPLEMENTATION

Earlier chapters dealt with the theoretical concepts required for the design of the project. In this chapter, the design and implementation of the project is discussed. The following sections give software requirements, overall working of this project and hardware design of the project.

3.1 Software Requisites

- **MATLAB**

The algorithm for the project is designed by using **MATLAB 8.2 release name R2013b**. **MATLAB** (**mat**rix **lab**oratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java,Fortran and Python.

The MATLAB environment (on most computer systems) consists of menus, buttons and a writing area similar to an ordinary word processor. There are plenty of help functions that you are encouraged to use. The writing area that you will see when you start MATLAB, is called the command window. In this window you give the commands to MATLAB. For example, when you want to run a program you have written for MATLAB you start the program in the command window by typing its name at the prompt. The command window is also useful if you just want to use MATLAB as a scientific calculator or as a graphing tool. If you write longer programs, you will find it more convenient to write the program code in a separate window, and then run it in the command window

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems.

Matlab has many toolboxes such as Financial Instruments Toolbox, Image Processing Toolbox, Neural Network Toolbox, and Statistical Toolbox. Machine Learning Toolbox, Data Acquisition Toolbox, DSP System Toolbox and so on. For our project, we need **Image Processing Toolbox** for image processing part and **Statistical and Machine Learning Toolbox** for SVM classifier.

➢ Software dependencies in Raspberry Pi 2

- **Raspbian Wheezy**

Raspbian is a free operating system based on Debian optimized for the Raspberry Pi hardware. An operating system is the set of basic programs and utilities that makes the Raspberry Pi run. However, Raspbian provides more than a pure OS: it comes with over 35,000 packages, pre-compiled software bundled in a nice format for easy installation on Raspberry Pi.

Raspbian is an unofficial port of Debian Wheezy armhf with compilation settings adjusted to produce optimized "hard float" code that will run on the Raspberry Pi. This provides significantly faster performance for applications that make heavy use of floating point arithmetic operations. All other applications will also gain some performance through the use of advanced instructions of the ARMv6 CPU in Raspberry Pi.

- **OpenCV**

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. OpenCV makes it easy for businesses to utilize and modify the code.

OpenCV is released under a BSD license and hence it’s free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform. Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 9 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics.

- **Python**

Python is a widely used general-purpose, high-level programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or Java. The language provides constructs intended to enable clear programs on both a small and large scale. Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.

Python interpreters are available for installation on many operating systems, allowing Python code execution on a wide variety of systems. Using third-party tools, such as Py2exe or Pyinstaller, Python code can be packaged into stand-alone executable programs for some of the most popular operating systems, allowing for the distribution of Python-based software for use on those environments without requiring the installation of a Python interpreter.

- **Scikit-learn**

Scikit-learn (formerly scikits.learn) is an open source machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, logistic regression, naive Bayes, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn is largely written in Python, with some core algorithms written in Cython to achieve performance.

- **Pip**

Pip is a package management system used to install and manage software packages written in Python. Many packages can be found in the Python Package Index (PyPI).

- **Numpy**

NumPy is an extension to the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays. NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and FORTRAN code
- Useful linear algebra, Fourier transform, and random number capabilities

▪ **Matplotlib**

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like wxPython, Qt, or GTK+. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB. SciPy makes use of matplotlib, and is distributed under a BSD-style license.

▪ **Skimage (Image Processing SciKit (Toolbox for SciPy))**

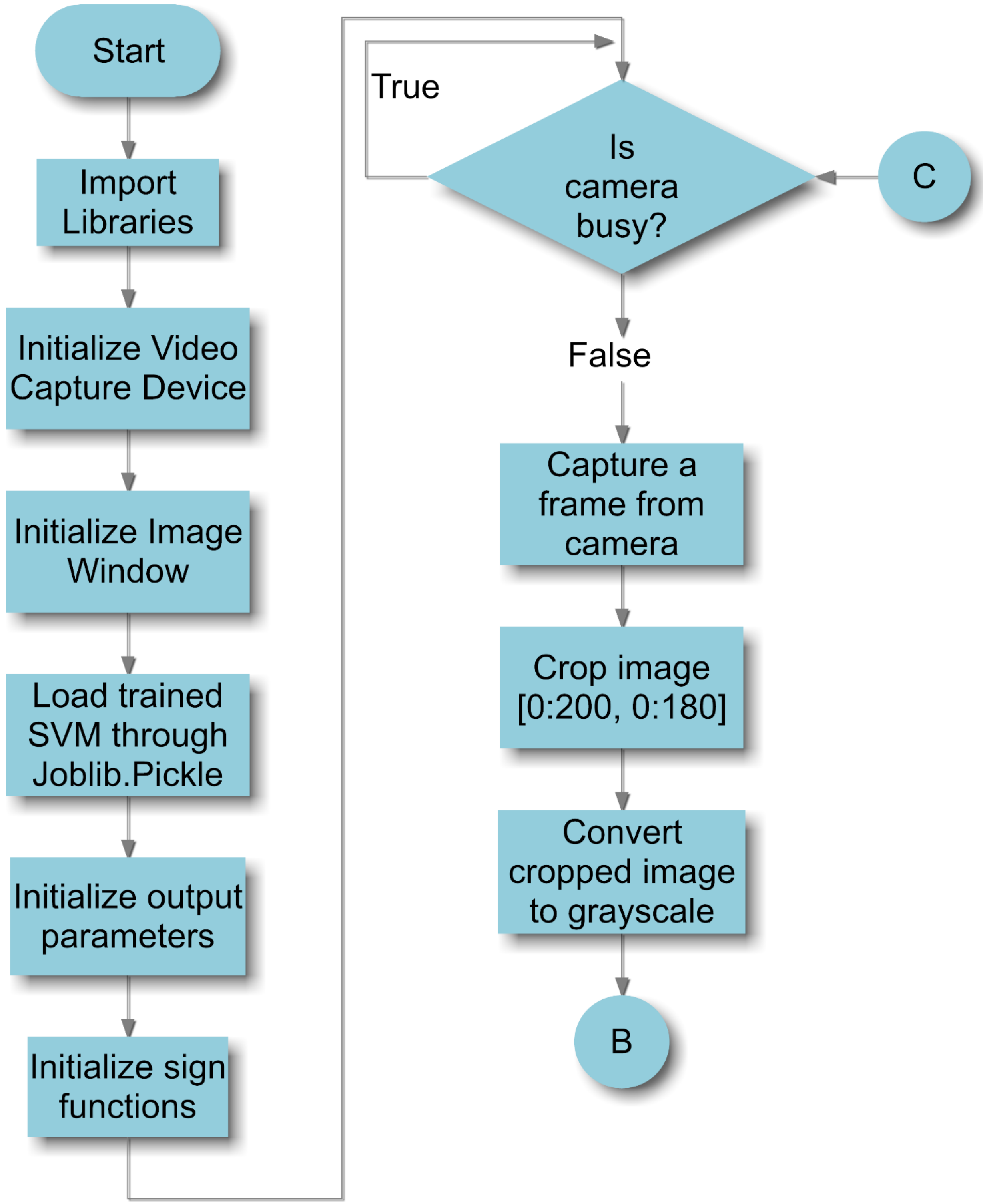
Scikit-image (also known as skimage) is a collection of algorithms for image processing and computer vision. The main package of skimage only provides a few utilities for converting between image data types; for most features, you need to import one of the following subpackages.

Subpackages:

- **Color:** Color space conversion.
- **Data:** Test images and example data.
- **Draw:** Drawing primitives (lines, text, etc.) that operate on NumPy arrays.
- **Exposure:** Image intensity adjustment, e.g., histogram equalization, etc.
- **Feature:** Feature detection and extraction, e.g., texture analysis corners, etc.
- **Filters:** Sharpening, edge finding, rank filters, thresholding, etc.
- **Graph:** Graph-theoretic operations, e.g., shortest paths.
- **Io:** Reading, saving, and displaying images and video.
- **Measure:** Measurement of image properties, e.g., similarity and contours.
- **Morphology:** Morphological operations, e.g., opening or skeletonization.
- **Novice:** Simplified interface for teaching purposes.

- **Restoration:** Restoration algorithms, e.g., deconvolution algorithms, denoising, etc.
- **Segmentation:** Partitioning an image into multiple regions.
- **Transform:** Geometric and other transforms, e.g., rotation or the Radon transform.
- **Util:** Generic utilities.
- **Viewer:** A simple GUI for visualizing results and exploring parameters.

3.2 System Flowchart



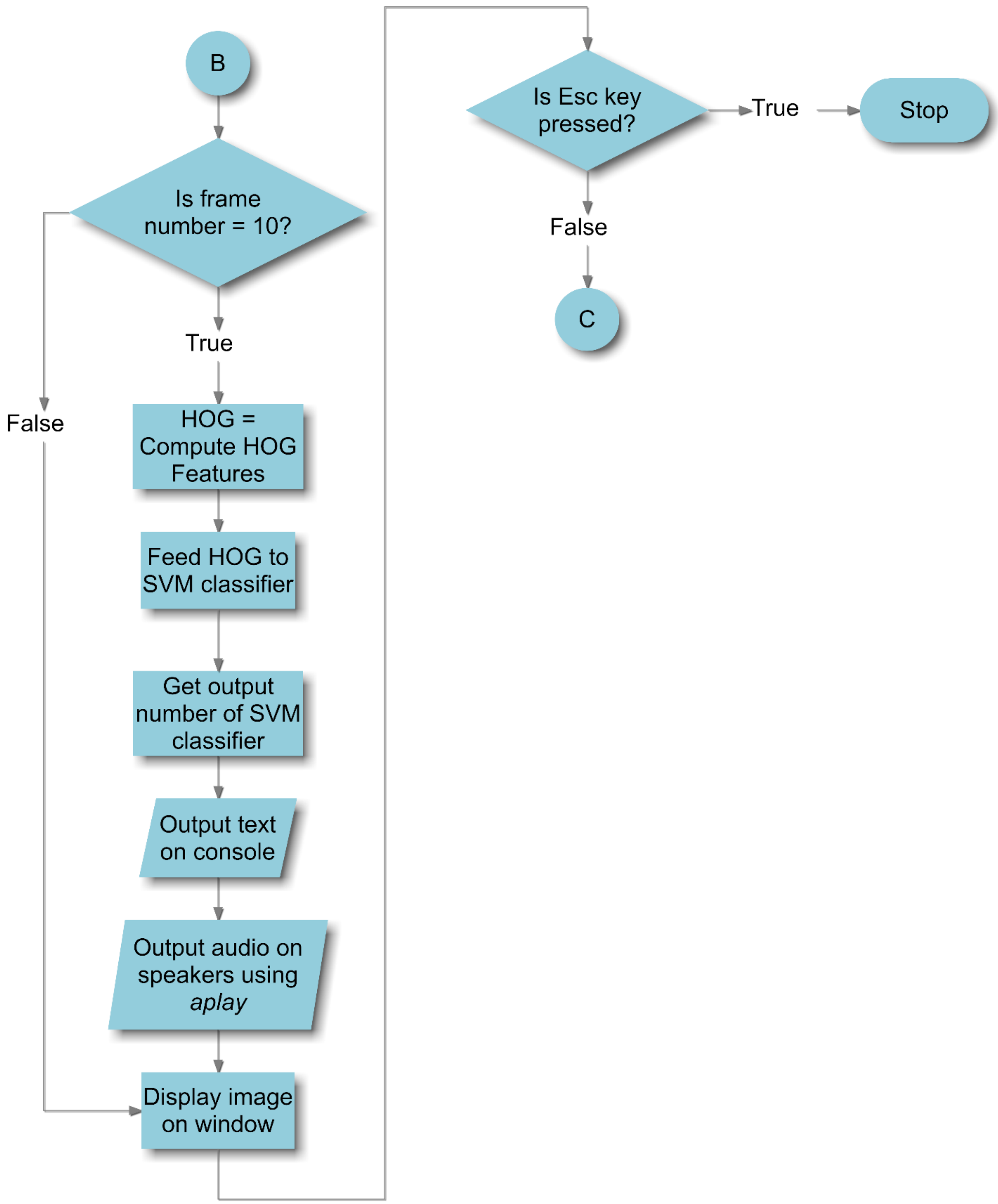


Figure 3.1: Translating Signs in real-time video feed.

3.2.1 Flowchart for taking images

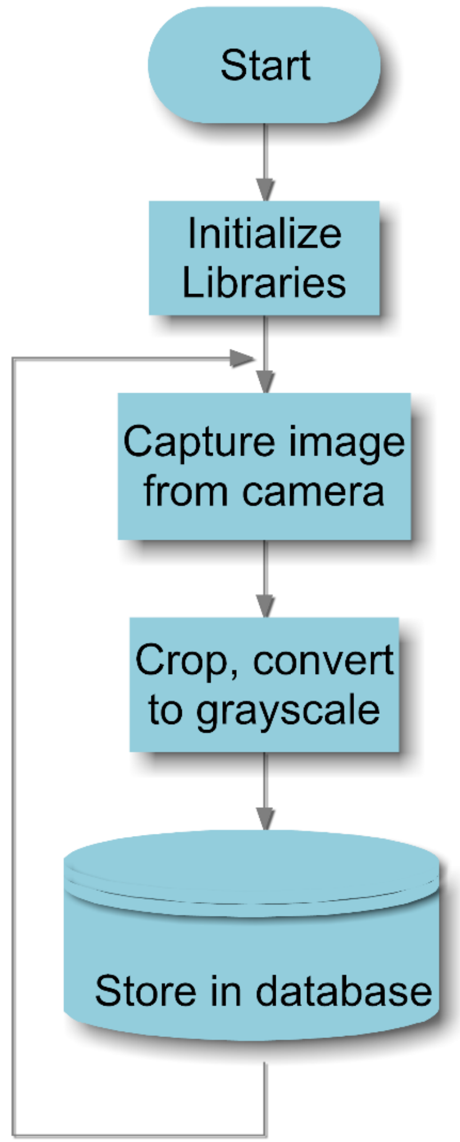


Figure 3.2: Flowchart for taking images and storing it.

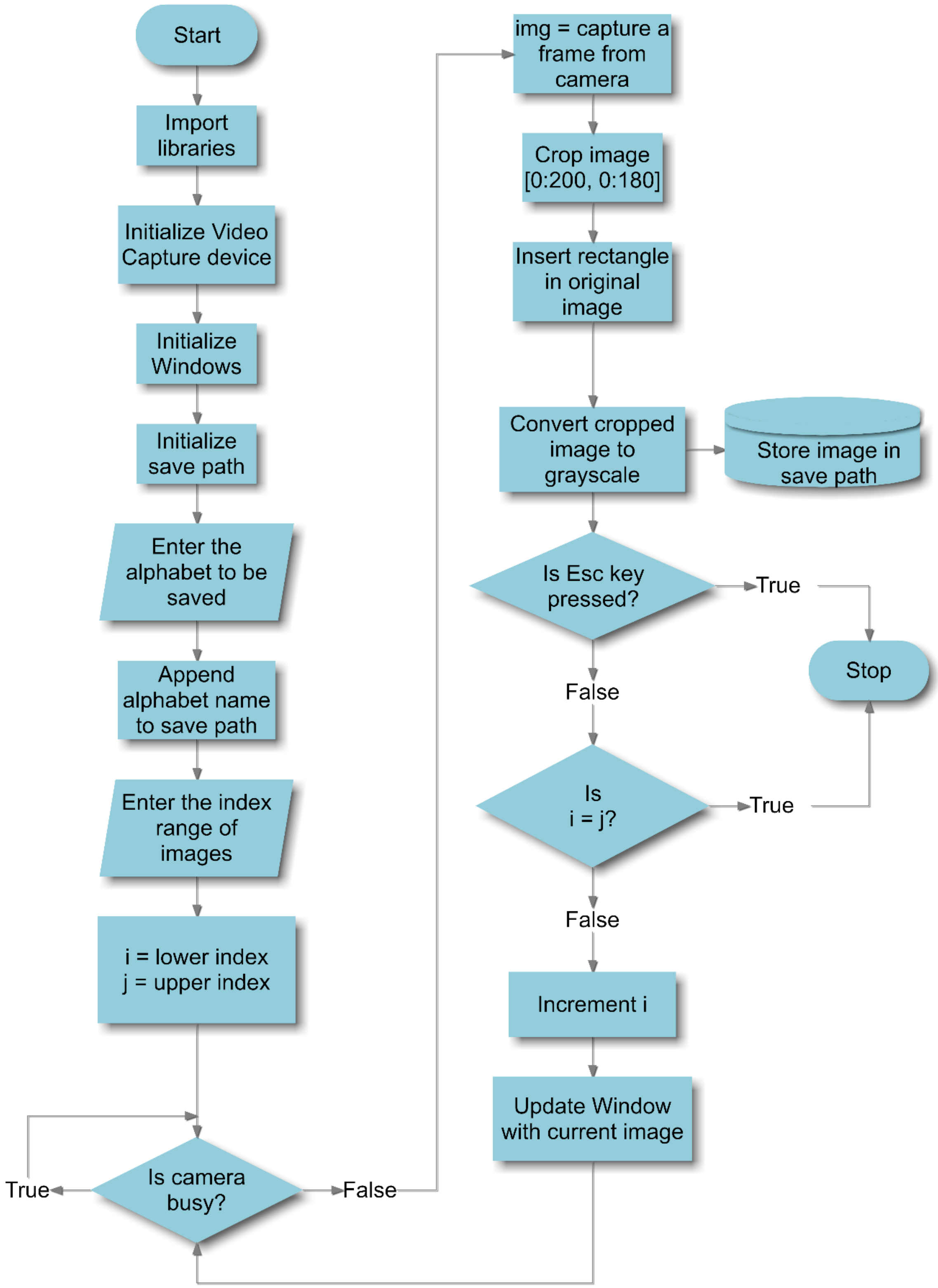


Figure 3.3: Taking image samples from camera and store in database.

3.2.2 Flowchart for Extracting HOG Features

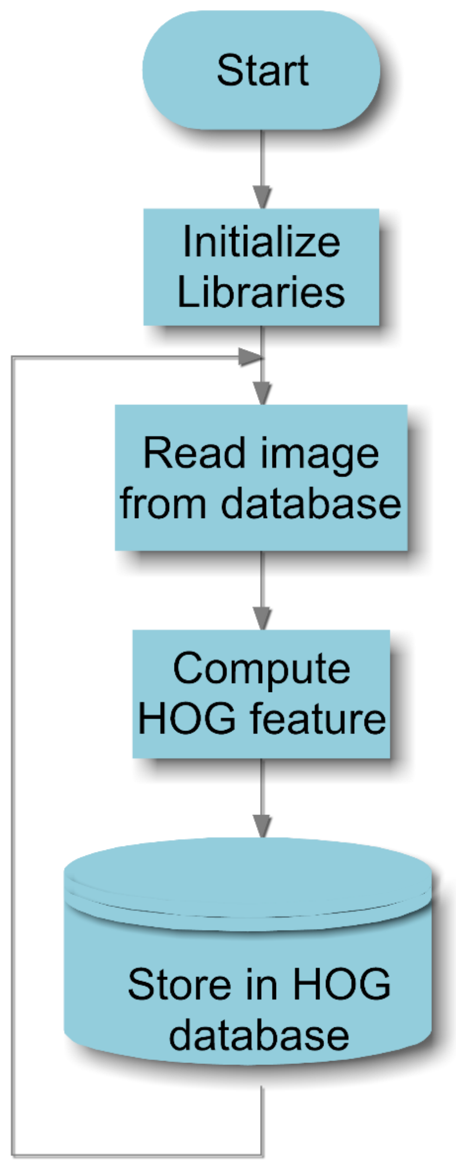


Figure 3.4: Computing HOG features.

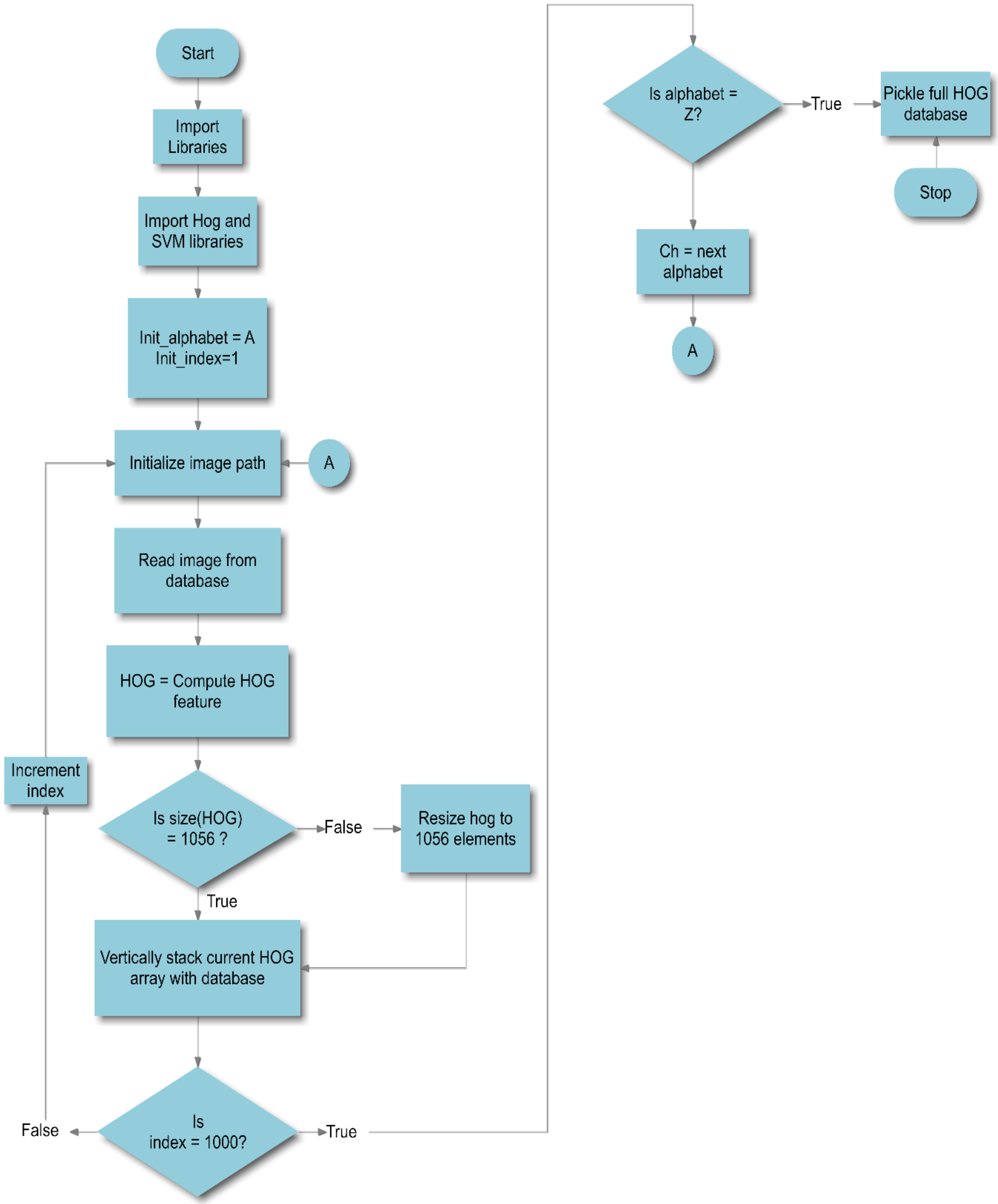
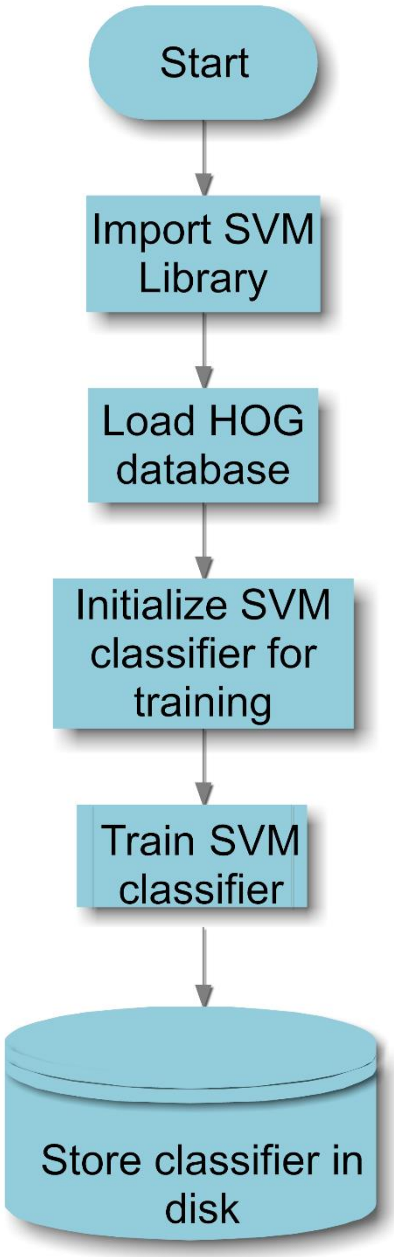
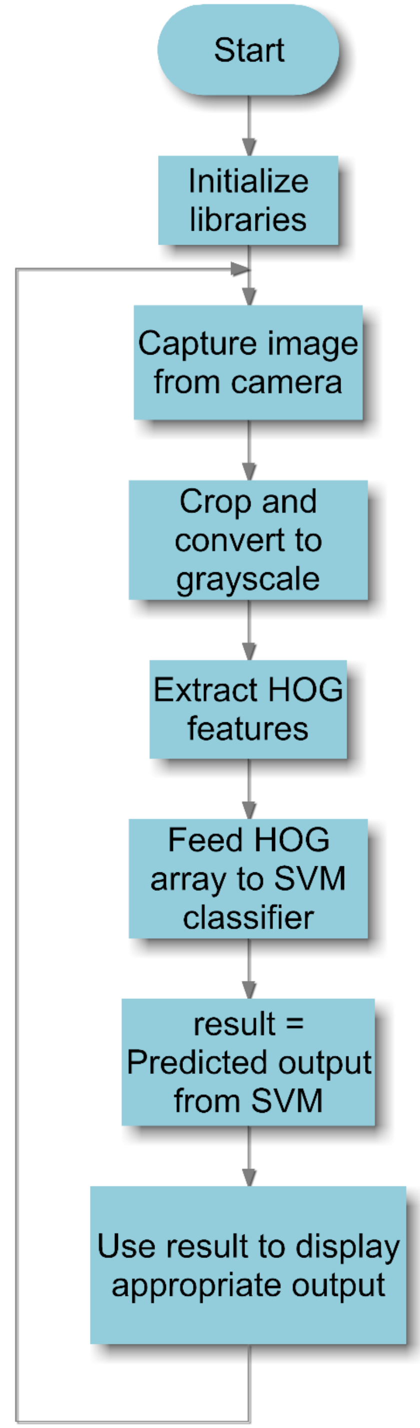


Figure 3.5: Computing HOG feature of all the images in database and generate update to HOG database.

3.2.3 Flowchart for SVM Training and Classification



(a)



(b)

Figure 3.6: Flowchart for (a) Training SVM classifier (b) Using SVM in real-time video

3.3 Hardware Design

In this section, a detailed explanation of all hardware part of the project is discussed.

3.3.1 Raspberry Pi 2

The Raspberry Pi is a series of credit card-sized single-board computers developed in the UK by the Raspberry Pi Foundation with the intention of promoting the teaching of basic computer science. The original Raspberry Pi and Raspberry Pi 2 are manufactured in several board configurations through licensed manufacturing agreements with Newark element14 (Premier Farnell), RS Components and Egoman.

The original Raspberry Pi is based on the Broadcom BCM2835 system on a chip (SoC), which includes an ARM1176JZF-S 700 MHz processor, VideoCore IV GPU, and was originally shipped with 256 megabytes of RAM, later upgraded (models B and B+) to 512 MB.[12] The system has Secure Digital (SD) (models A and B) or MicroSD sockets for boot media and persistent storage.

On 2nd February 2015, the next-generation Raspberry Pi, Raspberry Pi 2, was officially announced. Raspberry Pi 2 includes **ARM Cortex-A7 900MHz quad-core Broadcom BCM2836 CPU with 1GB DDR2** RAM. In this section, we discuss the hardware specifications of Pi which is associated with our project. Appendix B has the mechanical drawings of Raspberry Pi 2. [11]

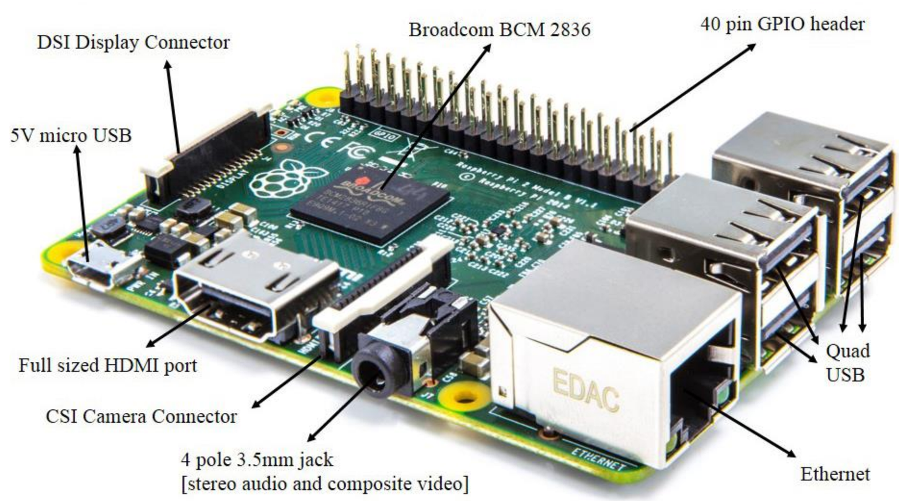


Figure 3.7: Raspberry Pi 2.

Technical Specifications:

- Broadcom BCM2836 Arm7 Quad Core Processor powered Single Board Computer running at 900MHz
- 1GB RAM
- 40 pin extended GPIO
- 4 x USB 2 ports
- 4 pole Stereo output and Composite video port
- Full size HDMI
- CSI camera port for connecting the Raspberry Pi camera
- DSI display port for connecting the Raspberry Pi touch screen display
- Micro SD port for loading your operating system and storing data
- Micro USB power source

One powerful feature of the Raspberry Pi is the row of GPIO (general purpose input/output) pins along the edge of the board, next to the yellow video out socket. These pins are a physical interface between the Pi and the outside world. At the simplest level, you can think of them as switches that you can turn on or off (input) or that the Pi can turn on or off (output). Seventeen of the 26 pins are GPIO pins; the others are power or ground pins as shown in figure 3.5.

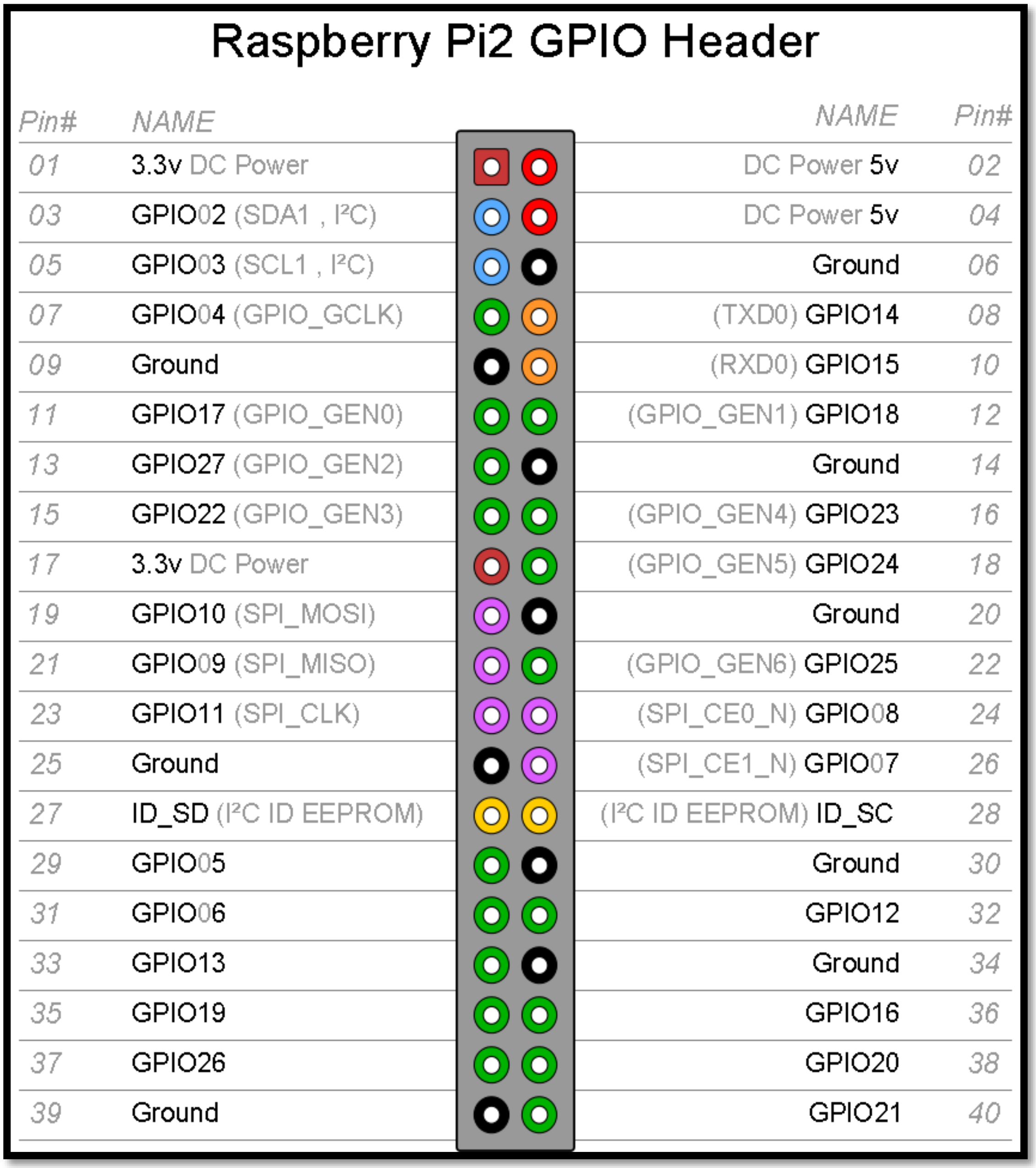


Figure 3.8: Raspberry Pi 2 GPIO Pin Configuration.

Functional Block Diagram

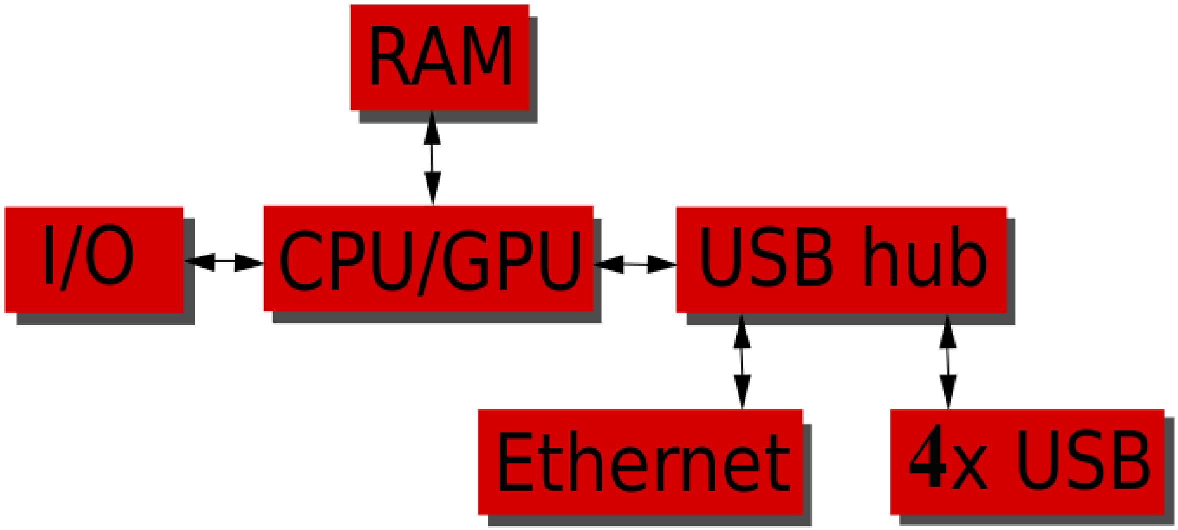


Figure 3.9: Raspberry Pi 2 Functional Block Diagram.

- **Power Supply**

The device is powered by a 5V micro USB supply. Exactly how much current (mA) the Raspberry Pi requires is dependent on what is to be connected to it. It has been found that purchasing a 1.2A (1200mA) power supply from a reputable retailer will provide us with ample power to run the Raspberry Pi.

The USB ports on a Raspberry Pi have a design loading of 100mA each - sufficient to drive "low-power" devices such as mice and keyboards. Devices such as WiFi adapters, USB hard drives, USB pen drives all consume much more current and should be powered from an external hub with its own power supply. While it is possible to plug a 500mA device into a Pi and have it work with a sufficiently powerful supply, reliable operation is not guaranteed.

- **Processor**

The Raspberry Pi is based on the Broadcom BCM2835 system on a chip (SoC), which includes 900 MHz quad core ARM Cortex-A7processor, VideoCore IV GPU, and RAM. It has a Level 1 cache of 16 KB and a Level 2 cache of 128 KB. The Level 2 cache is used primarily by the GPU. The SoC is stacked underneath the RAM chip, so only its edge is visible.

- **RAM**

Raspberry Pi 2 comes with 1GB DDR2 RAM (shared with GPU). It is sufficient for standalone 1080p video decoding, or for simple 3D, but not for both together.

- **Networking**

Raspberry Pi 2 can be connected to a network using Ethernet and Wi-Fi Adapter. The Ethernet port is provided by a built-in USB Ethernet adapter.

- **GPIO**

General Purpose Input Output pins can be configured as either general-purpose input, general-purpose output or as one of up to 6 special alternate settings, the functions of which are pin-dependent. The pads are configurable CMOS push-pull output drivers/input buffers. Register-based control settings are available for

  - Internal pull-up / pull-down enable/disable
  - Output drive-strength
  - Input Schmitt-trigger filtering

- **Peripherals**

Generic USB keyboards and mice are compatible with the Raspberry Pi.

- **USB**

The Raspberry Pi 2 is equipped with four USB2.0 ports. These are connected to the on-board 5-port USB hub. The USB ports enable the attachment of peripherals such as keyboards, mice, webcams that provide the Pi with additional functionality.

- **Port Power Limits**

USB devices have defined power requirements, in units of 100mA from 100mA to 500mA. The device advertises its own power requirements to the USB host when it is first connected. In theory, the actual power consumed by the device should not exceed its stated requirement.

The USB ports on a Raspberry Pi have a design loading of 100mA each - sufficient to drive "low-power" devices such as mice and keyboards. Devices such as WiFi adapters, USB hard drives, USB pen drives all consume much more current and should be powered from an external hub with its own power supply. While it is possible to plug a 500mA device into a Pi and have it work with a sufficiently powerful supply, reliable operation is not guaranteed.

Camera Serial Interface [CSI]

The **CSI connector** consists of two smaller interfaces. The first interface is for the transfer of data and clock signals from the camera to the processor in one direction only. The second interface consists of SCL / SDA lines, which is a bidirectional control link.

The two data lanes on the CSI-2 bus provide a theoretical 2 Gbps bandwidth, which approximates to around 5 MP resolution. It is very likely to have a maximum video recording resolution of 1920 pixels × 1080 pixels at around 30 frames per second.

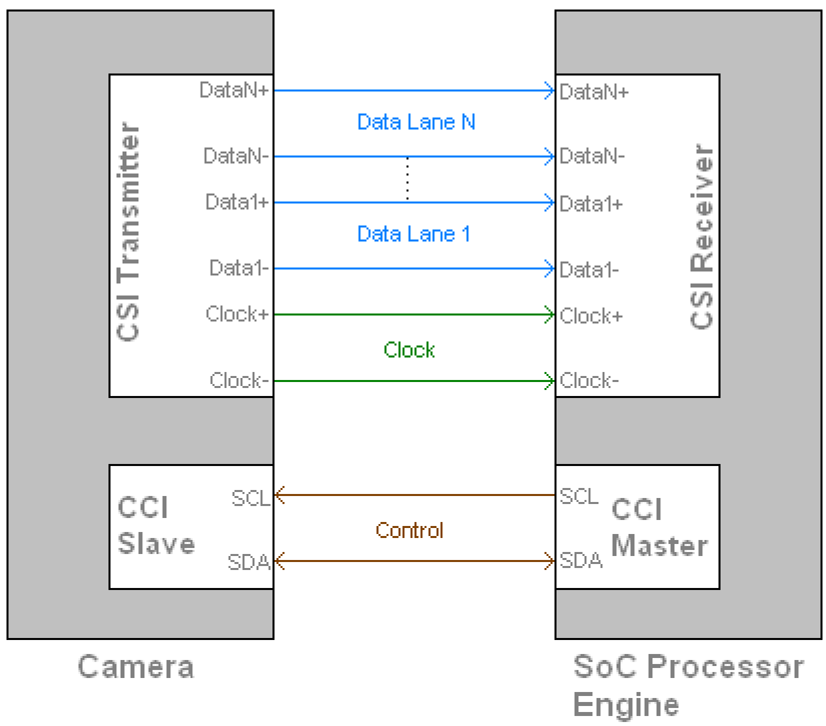


Figure 3.10: Raspberry Pi CSI-2 Connector Specifications.

The Raspberry Pi CSI connector is a surface mount ZIF 15 socket, used for interfacing a camera through a ribbon cable. The CSI-2 specifications are very detailed and describe the physical layer known as D-PHY2. The signaling scheme of this physical layer, known **as Low Voltage**

Differential Signalling (SubLVDS). It is a system for low voltage 1.2 V applications, allowing data rates of up to 800 Mbps per lane with 1 Gbps set as a practical limit. In practice, the data rate can vary a lot and depends upon the quality of the interconnections. A maximum of four physical data lanes are allowable in this specification, however two are available for the Raspberry Pi.

This is a high-speed data communication bus and noise is of huge concern to the design engineer. Although this type of serial communication generates **negligible crosstalk**, the specification suggests using **minimum clock rates** for the camera module. The CSI transmission clock is source synchronous and the main processor may produce it instead to avoid noise interference on the camera module. The data transmission supports a wide range of data types such as RGB, RAW, YUV, generic, or byte based programmer defined.

Display Serial Interface [DSI]

It is a specification by the Mobile Industry Processor Interface (MIPI) Alliance aimed at reducing the cost of display controllers in a mobile device. It is commonly targeted at LCD and similar display technologies. It defines a serial bus and a communication protocol between the host (source of the image data) and the device (destination of the image data).

At the physical layer, DSI specifies a high-speed differential signaling point-to-point serial bus. This bus includes one high speed clock lane and one or more data lanes. Each lane is carried on two wires (due to differential signaling). All lanes travel from the DSI host to the DSI device, except for the first data lane (lane 0), which is capable of a bus turnaround (BTA) operation that allows it to reverse transmission direction.

HDMI (High-Definition Multimedia Interface)

It is a proprietary audio/video interface for transferring uncompressed videodata and compressed or uncompressed digital audio data from an HDMI-compliant source device, such as a display controller, to a compatible computer monitor, video projector, digital television, or digital audio device. HDMI is a digital replacement for analog video standards.

3.3.2 Pi Camera

The Raspberry Pi camera module is capable of taking full HD 1080p photo and video and can be controlled programmatically. The specifications of the camera is listed in table 3.1. Hardware and software features of pi camera a listed in appendix C.

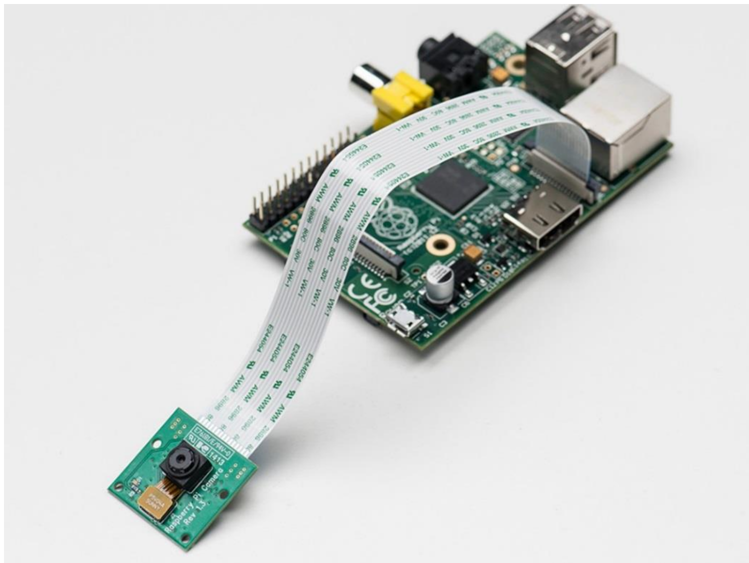


Figure 3.11: Pi Camera.

The Raspberry Pi camera module transfers data through an extremely fast **Camera Serial Interface (CSI-2)** bus directly to the Broadcom BCM2836 system-on-chip (SoC) processor. It does this through a 15-pin ribbon cable, also known as a **flex cable**, and connects to the surface mount ZIF 15 socket.

Figure 3.12: Comparison of size of camera with size of a British penny.

Many cameras follow the MIPI standard. The principle of serialising CCD data is the same and there are only a few manufacturers of these chips. These camera modules can be very small; this one is smaller than a British penny as shown in figure 3.9.

The technical specification of Pi camera is in table 3.1.

Size	25 x 20 x 9 mm
Weight	3 g
Still resolution	5 Megapixels
Video modes	1080p30, 720p60 and 640x480p60/90
Linux integration	V4L2 driver available
C programming API	OpenMAX IL and others available
Sensor	OmniVision OV5647
Sensor resolution	2592 x 1944 pixels
Sensor image area	3.76 x 2.74 mm
Pixel size	1.4 µm x 1.4 µm
Optical size	1/4"
Full-frame SLR lens equivalent	35 mm
S/N ratio	36 dB
Dynamic range	67 dB @ 8x gain
Sensitivity	680 mV/lux-sec
Dark current	16 mV/sec @ 60 C
Well capacity	4.3 Ke-

Table 3.1: Technical Specifications of Pi Camera.

3.3.3 NTSC TFT-LCD Color Monitor

NTSC stands for **National Television System Committee**. It is the analog television system that was used in most of the Americas. It uses a luminance-chrominance encoding system. In NTSC, chrominance is encoded using two color signals known as I (in-phase) and Q (in quadrature) in a process called QAM.

A **thin film transistor liquid crystal display (TFT-LCD)** is a technology which is used in LCD monitor. TFT technology can be used to give one of the clearest pictures of any flat screen display and it uses much less electricity than older screens. TFT displays are very fragile because they are made as thin and light as possible but this means they need far less space than the older CRT displays. Working of TFT-LCD is in appendix D.

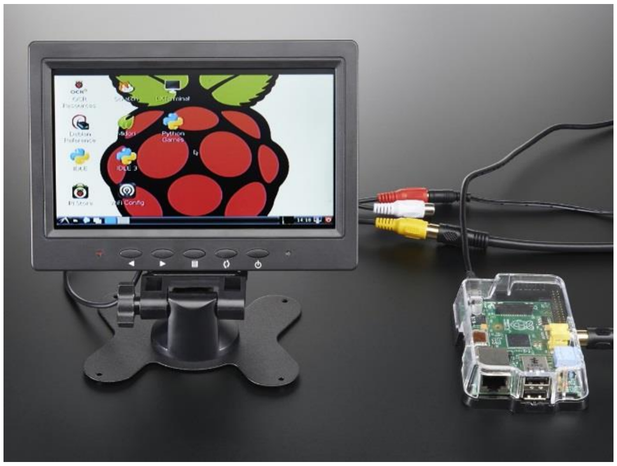


Figure 3.13: NTSC TFT-LCD 7´´ display connected to a Raspberry Pi.

Technical specifications of the display used is as follows:

- Power with 9-12VDC only
- Visible display dimensions: 154.5mm x 86.8mm
- Selectable 16:9 or 4:3 ratio via menu
- Resolution: 480 x RGB x 234
- Brightness: 250cd/m2
- Contrast: 400:1
- Display plastic case dimensions: 172mm x 114.28mm
- Weight (excluding power cable): 383.2g

3.3.4 RCA Connector

The name "RCA" derives from the **Radio Corporation of America**, which introduced the design by the early 1940s for internal connection of the pickup to the chassis in home radio-phonograph consoles. An RCA connector, sometimes called a phono connector or cinch connector, is a type of electrical connector commonly used to carry audio and video signals. The connectors are also sometimes casually referred to as A/V jacks.

The connection's plug is called an RCA plug or phono plug, for "phonograph." The name "phono plug" is sometimes confused with a "phone plug" which may refer to a quarter-inch "phone plug". RCA connector is used while interfacing Raspberry Pi 2 with TFT-LCD display.

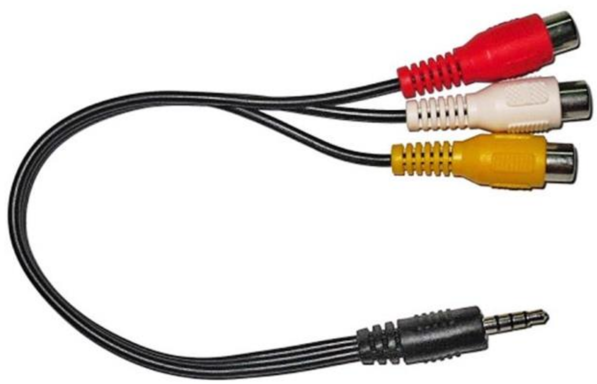


Figure 3.14: 4 pole 3.5mm RCA connector.

4 Pole 3.5mm Connectors

A phone connector is a common family of connector typically used for analog signals, primarily audio. It is cylindrical in shape, typically with two, three or four contacts. Three-contact versions are known as TRS connectors, where T stands for "tip", R stands for "ring" and S stands for "sleeve". Similarly, two- and four-contact versions are called TS and TRRS connectors respectively.

The "mini" connector has a diameter of 3.5 mm (approx. 1⁄8 inch). It is also termed an audio jack, phone jack, phone plug, and jack plug. Specific models are termed stereo plug, mini-stereo, mini jack, headphone jack, microphone jack, tiny telephone connector, bantam plug.

3.3.5 TP-LINK Wi-Fi Adapter

Wi-Fi adapters provide connectivity to the local area network (LAN) in the home or office. Typically used to add Wi-Fi to desktop computers, they can also retrofit older laptops that never came with Wi-Fi. TP Link Wi-Fi adapter gives a wireless speed up to 150 Mbps. It supports Ad-Hoc and Infrastructure modes. Has a USB 2.0 connector. Also supports Roaming Technology with an efficient Security Setup Button. It gives speedy wireless transmission at up to 150Mbps ideal for video streaming or internet calls.

3.3.6 Logitech Wireless Keyboard and Mouse

It gives an advanced 2.4 GHz wireless connection which virtually eliminates delays, dropouts and interference and gives a range of up to 10 meters. Also has 128-bit AES encryption that helps protect information by encrypting data transfer between the keyboard and receiver with one of the highest levels of data encryption available. Gives a 24 month keyboard battery life with auto standby that save energy. Also eight hot keys for instant access to the Internet, e-mail, play/pause, volume etc. It has Plug-and-play feature that is installed without software. Compact keyboard with durable, UV-coated keys and finger grooves on top of the mouse with 1000 dpi optical tracking.

3.4 Summary

The topics presented in this chapter provides the details of implementation of the project with the necessary flowcharts along with software and hardware requirements. The next chapter provides the implementation results followed by the discussions on the results obtained.

CHAPTER 4

RESULT AND DISCUSSIONS

An Image Processing Technique For The Translation Of Indian Sign Language Finger-Spelling To Audio And Text is implemented using the Raspberry Pi 2. The algorithm for the translation is developed using MATLAB 8.2 release name R2013b on Intel® i3 core® CPU @ 2.66GHz processor machine, Windows 8 (64 bit), 4GB RAM and a webcam with resolution of 640x480.

This project is a specific application project which is meant for translation of sign language. The translation of other sign languages can be implemented on the code programmed by changing the training samples of classifier.

4.1 Results Analysis

The prototype has been built and tested in particular lighting condition. The prototype recognizes the signs, displays the alphabet on TFT-LCD and plays appropriate audio file.

Data set and Parameters considered

The data set used for the training consists 1000 samples of 24 signs of ISL. A total of 2400 images are stored in database. The HOG features extracted for all images forms a matrix of 24000x1056. We tested the system with live video stream and achieved a good success.

Results

Table 4.1 describes confusion matrix obtained using SVM classification technique for few images. Similar procedure is carried out for other video frames. The overall recognition rate is calculated and found to be 94.25%. Alphabets ‘M’ ‘N’ and ‘S’ have confusion between each other.

	A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
A	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	73	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	25	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
O	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	98	2	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	100	0	0	0	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	82	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25

Table 4.1: Confusion matrix for SVM classification technique.

4.2 Summary

This chapter presented the results of implementation of the translation of ISL. The different aspects of results has been discussed and found to be efficient. The next chapter provides the conclusion and future scope of this project.

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENT

“A gift consists not in what is done or given, but in the intention of the giver”. These lines of Seneca, inspired us to bring in a change for many hearing impaired people. Our project, “An Image Processing Technique For The Translation Of Indian Sign Language Finger-Spelling To Audio And Text”, aims to extend help for hearing impaired people to communicate.

In this project we developed a translation system using Raspberry Pi 2. This prototype is designed to be a standalone device translating ISL to audio and text. This project enables easily communication for the hearing impaired people .However, the limitation with this project is to maintain constant lighting condition.

The techniques used, HOG Feature Extraction and SVM classifier, are helpful in translation of ISL. This approach for translation of ISL is novel as per our knowledge. The use of Raspberry Pi 2, Pi Camera and TFT-LCD display has made the project a reliable standalone device.

5.1 Advantages

- Effective translation of Sign languages.
- This system helps to mediate between people using other sign languages.

5.2 Disadvantages

- Cost is high.
- Time delay.
- To drive the display Raspberry Pi 2 needs 5V and 3A.

5.3 Future Enhancement

The various advantages are observed in the last section. With this blueprint, it can be extended to several applications.

The project can also be extended as

- More features can be extracted for signs ‘M’, ‘N’ and “S’ for better efficiency.
- We can implement translation for other sign languages.
- Android application can be developed by using this approach.

5.4 Summary

This chapter concludes with the different advantages and disadvantages of the project. It also provides the scope for the future work which could be carried out on translation of ISL to audio and text.

REFERENCES

1. Meir, Sandler, Padden, & Aronoff, "Emerging sign languages." In Marschark & Spencer, eds., Oxford Handbook of Deaf Studies, Language, and Education.
2. Erard, Michael (July 19, 2005). "How Linguists and Missionaries Share a Bible of 6,912 Languages". The New York Times.
3. Bauman, Dirksen (2008). Open your eyes: Deaf studies talking. University of Minnesota Press. ISBN 0-8166-4619-8.
4. Lewis, M. Paul; Simons, Gary F.; Fennig, Charles D., eds. (2013)."Ethnologue: Languages of the World" (17th ed.). SIL International. Retrieved 2013-12-03.
5. http://indiansignlanguage.org/
6. ttp://indiansignlanguage.org/isl-dictionary/alphabets/double-handed
7. http://indiansignlanguage.org/isl-dictionary/alphabets/single-handed
8. Rafael C Gonzalez and Richard E Woods, "Digital Image Processing", third edition, Pearson Publications.
9. Dalal, N. and Triggs, B., "Histograms of Oriented Gradients for Human Detection," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, San Diego, CA, USA.
10. Pooja Kamavisdar, Sonam Saluja, Sonu Agrawal "A Survey on Image Classification Approaches and Techniques", International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 1, January 2013.
11. http://scikit-learn.org/stable/modules/svm.html
12. https://www.raspberrypi.org/documentation/
13. http://www.element14.com/community/docs/
14. http://www.tomshardware.com/reviews/tft-guide,114-4.html
15. Stephan Liwicki, Mark Everingham, "Automatic Recognition of Fingerspelled Words in British Sign Language"
16. Ghotkar, A. S., Khatal, R., Khupase, S., Asati, S. and Hadap, M. "Hand Gesture Recognition for Indian Sign Language", International Conference on Computer Communication and Informatics (ICCCI), Jan. 10-12, 2012, Coimbatore, India.

17. Starner, T. and Pentland, A. "Real-Time American Sign Language Recognition from Video Using Hidden Markov Models", IEEE, 1995, pp. 265-270.Singha, J. and Das, K. "Hand Gesture Recognition Based on Karhunen-Loeve Transform", Mobile and Embedded Technology International Conference (MECON), January 17-18, 2013, India, pp. 365-371.
18. Singha, J. and Das, K. "Indian Sign Language Recognition Using Eigen Value Weighted Euclidean Distance Based Classification Technique", International Journal of Advanced Computer Science and Applications, Vol. 4, No. 2, 2013, pp. 188-195.

APPENDIX A

Computation of hyperplane in SVM

Let's introduce the notation used to define formally a hyperplane:

$$\mathbf{f(x) = \beta_0 + \beta^T x}$$

Where β is known as the weight vector and β_0 as the bias.

The optimal hyperplane can be represented in an infinite number of different ways by scaling of β and β_0. As a matter of convention, among all the possible representations of the hyperplane, the one chosen is

$$| \beta_0 + \beta^T x | = 1$$

Where x symbolizes the training examples closest to the hyperplane. In general, the training examples that are closest to the hyperplane are called support vectors. This representation is known as the canonical hyperplane.

Now, we use the result of geometry that gives the distance between a point x and a hyperplane(β,β_0):

$$\text{Distance} = \{ | \beta_0 + \beta^T x | \} \ / \ \{ ||\beta|| \}$$

In particular, for the canonical hyperplane, the numerator is equal to one and the distance to the support vectors is

$$\text{Distance}_{\text{support vectors}} = \{ | \beta_0 + \beta^T x | \} \ / \ \{ ||\beta|| \} = 1 /\{ ||\beta|| \}$$

The margin introduced in the previous section, here denoted as M, is twice the distance to the closest examples:

$$M = 2 / \{||\beta||\}$$

Finally, the problem of maximizing M is equivalent to the problem of minimizing a function L(β) subject to some constraints. The constraints model the requirement for the hyperplane to classify correctly all the training examples x_i. Formally,

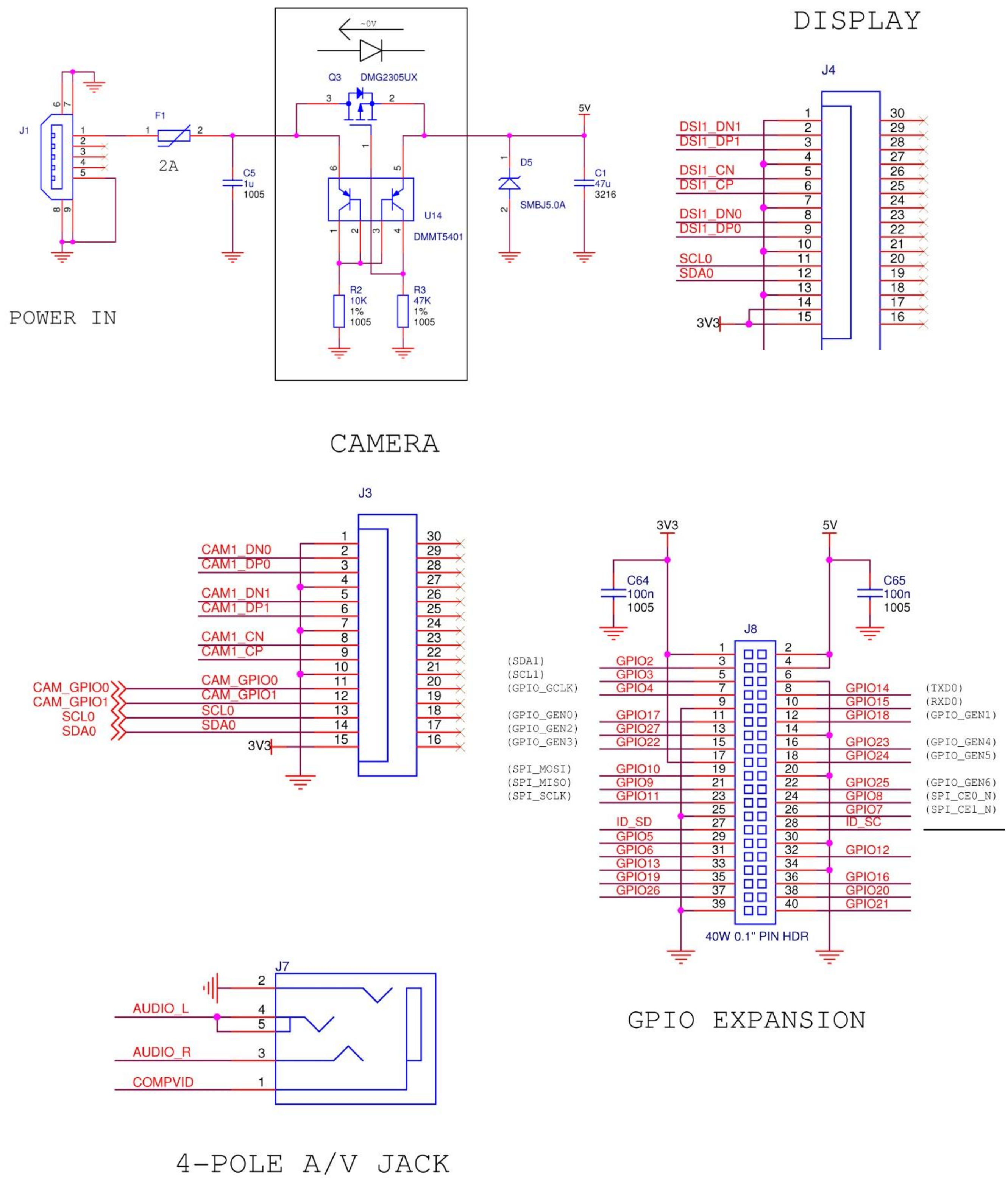
$$\min\{ \beta,\beta_0\} L(\beta) = ½ || \beta ||^2 \text{ subject to } y_i (\beta^T x_i + \beta_0) \geq 1 \ ¥ i$$

Where y_i represents each of the labels of the training examples.

This is a problem of Lagrangian optimization that can be solved using Lagrange multipliers to obtain the weight vector β and the bias β_0 of the optimal hyperplane.

APPENDIX B

Raspberry pi 2 Drawings



APPENDIX C

Hardware Specifications of Pi camera

Available	Implemented
Chief Ray Angle Correction	Yes
Global and rolling shutter	Rolling shutter
Automatic exposure control (AEC)	No - done by ISP instead
Automatic white balance (AWB)	No - done by ISP instead
Automatic black level calibration (ABLC)	No - done by ISP instead
Automatic 50/60 Hz luminance detection	No - done by ISP instead
Frame rate up to 120 fps	max 90fps. Limitations on frame size for the higher frame rates (VGA only for above 47fps)
AEC/AGC 16-zone size/position/weight control	No - done by ISP instead
Mirror and flip	Yes
Cropping	No - done by ISP instead (except 1080p mode)
Lens correction	No - done by ISP instead
Defective pixel canceling	No - done by ISP instead
10-bit RAW RGB data	Yes , format conversions available via GPU
Support for LED and flash strobe mode	LED flash
Support for internal and external frame synchronization for frame exposure mode	No

Software Specifications of Pi camera

Picture formats	JPEG (accelerated) , JPEG + RAW , GIF , BMP , PNG , YUV420 , RGB888
Video formats	raw h.264 (accelerated)
Effects	negative , solarise , posterize , whiteboard , blackboard , sketch , denoise , emboss , oilpaint , hatch , gpen , pastel , watercolour, film , blur , saturation
Exposure modes	auto , night , nightpreview , backlight , spotlight , sports , snow , beach , verylong , fixedfps , antishake , fireworks
Metering modes	average, spot, backlit, matrix
Automatic White Balance modes	off, auto , sun , cloud, shade, tungsten, fluorescent , incandescent , flash, horizon
Triggers	Keypress , UNIX signal , timeout
Extra modes	demo , burst/timelapse , circular buffer , video with motion vectors , segmented video , live preview on 3D models

APPENDIX D

Working of TFT-LCD Display

Liquid crystals can change their molecular structure and therefore allow varying levels of light to pass through them (or they can block the light). Two polarizer filters, color filters and two alignment layers determine exactly how much light is allowed to pass and which colors are created. The layers are positioned between the two glass panels. A specific voltage is applied to the alignment layer, creating an electric field - which then aligns the liquid crystals. Each dot on the screen (pixel) therefore requires three components, one for red, green and blue - just as for the tubes within cathode ray tube devices.

The most common devices are Twisted Nematic TFTs. The following sections explain the way in which such TFTs work. A number of different technologies obviously exist.

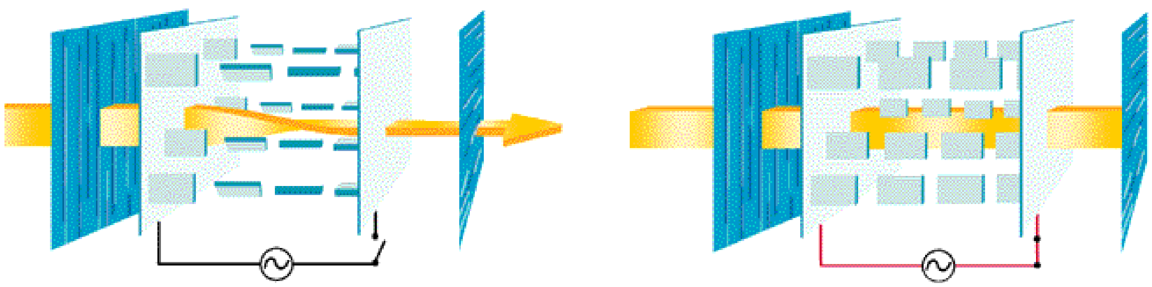


Figure: How a Standard TFT (twisted nematic) works.

When no voltage is applied, the molecule structures are in their natural state and twisted by 90 degrees. The light emitted by the back light can then pass through the structure. If a voltage is applied, i.e. an electric field is created, the liquid crystals are twisted so that they are vertically aligned. The polarized light is then absorbed by the second polarizer. Light can therefore not leave the TFT display at this location.