



**SIGN LANGUAGE TO TEXT AND
SPEECH TRANSLATION USING IMAGE
PROCESSING**



A PROJECT REPORT

Submitted by

ASHWINI.K

Reg. No.:1110107012

KEERTHANA.M

Reg. No.:1110107044

MARIA CATHERINE

Reg. No.:1110107051

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION

ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE-641049

(An Autonomous Institution Affiliated to Anna University, Chennai)

APRIL 2015

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE-641049

(An Autonomous Institution Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report titled “**SIGN LANGUAGE TO TEXT AND SPEECH TRANSLATION USING IMAGE PROCESSING**” is the bonafied work of “**ASHWINI.K, KEERTHANA.M, MARIA CATHERINE**” who carried out the project work under my supervision.

SIGNATURE

Ms. R. Dhivya Praba, M.E.,
Supervisor,
Assistant Professor-II, Department of ECE,
Kumaraguru College of Technology,
Coimbatore641049.

SIGNATURE

Dr.Rajeswari Mariappan, M.E., Ph.D
Head Of The Department,
Department of ECE,
Kumaraguru College of Technology,
Coimbatore641049.

The candidates with Register numbers 1110107012, 1110107044, and 1110107051 are examined by us in the project viva-voce examination held on

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

First we would like to express our praise and gratitude to the Lord, who has showered his grace and blessing enabling us to complete this project in an excellent manner. He has made all things in beautiful in his time.

We express our sincere thanks to our beloved Joint Correspondent, **Shri. Shankar Vanavarayar** for his kind support and for providing necessary facilities to carry out the project work.

We would like to express our sincere thanks to our beloved Principal **Dr.R.S. Kumar M.E., Ph.D.**, who encouraged us with his valuable thoughts.

We would like to express our sincere thanks and deep sense of gratitude to our HOD, **Dr.Rajeswari Mariappan, M.E., Ph.D.**, for her valuable suggestions and encouragement which paved way for the successful completion of the project.

We are greatly privileged to express our deep sense of gratitude to the Project Coordinator **Ms.S.Nagarathinam, M.E., (Ph.D)**, Assistant Professor (SRG),for her continuous support throughout the course.

In particular, We wish to thank and express our everlasting gratitude to the Supervisor **Ms.R.DhivyaPraba, M.E.**, Assistant Professor for her expert counselling in each and every steps of project work and we wish to convey our deep sense of gratitude to all teaching and non-teaching staff members of ECE Department for their help and cooperation.

Finally, we thank our parents and our family members for giving us the moral support in all of our activities and our dear friends who helped us to endure our difficult times with their unfailing support and warm wishes.

TABLE OF CONTENTS

CONTENTS

CHAPTER	TITLE	PAGE NO
	LIST OF ABBREVIATIONS	7
	LIST OF FIGURES	8
	ABSTRACT	9
1	INTRODUCTION	10
	1.1 HARDWARE APPROACH	11
	1.2 BLOCK DIAGRAM AND ITS DECRPTION	11
	1.2.1 POWER SUPPLY	12
	1.2.2 FLEX SENSOR	12
	1.2.3 PIC 16F877A MICROCONTROLLER	13
	1.2.4 VOICE RECORD AND PLAYBACK	13
	1.3 CIRCUIT DIAGRAM AND ITS DECRPTION	14
	1.3.1 POWER SUPPLY	15
	1.3.2 TRANSFORMER	15
	1.3.3 BRIDGE RECTIFIER	15
	1.3.4 FILTER	16
	1.3.5 IC VOLTAGE REGULATORS	16
	1.3.6 FLEX SENSOR	17
	1.3.7 AMPLIFIERS	17
	1.3.8 PIC 16F877A MICROCONTROLLER	18

	1.3.9 WTV-SR VOICE RECORDER	18
	1.4 WORKING PRINCIPLE	18
	1.5 DIADVANTAGES OF HARDWARE APPROACH	19
2	SOFTWARE APPROACH	20
	2.1 DIGITAL IMAGE PROCESSING	20
	2.2 CLASSIFICATION OF IMAGES	21
	2.2.1 BINARY IMAGE	21
	2.2.2 GRAY SCALE IMAGE	21
	2.2.3COLOR IMAGE	22
	2.3 MATLAB	23
	2.4 FLOWCHART AND ITS DECRPTION	25
	2.5 ALGORITHM	26
	2.6 WORKING PRINCIPLE	26
	2.7 IMAGE DATABASE	27
	2.8 LARGE OBJECT TRACKING	29
	2.9 SEGMENTATION	29
	2.9.1 REGION GROWING APPROACH	29
	2.9.2 DRAWBACKS	30
	2.10 BACKGROUND MODELLING	31
	2.10.1 PREPROCESSING	32
	2.10.2 BACKGROUND SUBTRACTION	32
	2.10.3 POST PROCESSING	33
	2.11 SHAPE RECOGNITION	33
	2.11.1 DRAWBACKS	34

	2.12 ADVANTAGES OF SOFTWARE APPROACH	34
3	METHODOLOGIES	35
	3.1 GAUSSIAN SMOOTHING PROCESS	35
	3.2 MATHEMATICAL MORPHOLOGY	36
	3.2.1 BINARY MORPHOLOGY	36
	3.3 CONNECTED COMPONENT ANALYSIS	37
	3.3.1 LOCAL REGION DESCRIPTOR	38
	3.3.2 TEMPLATE MATCHING	38
	3.4 CORRELATION COEFFICIENT	39
	3.4.1 CROSS CORRELATION	39
4	RESULTS	40
5	CONCLUSION AND FUTURE WORKS	44
6	APPENDIX	45
7	REFERENCE	53

LIST OF ABBREVIATION

BSL	British Sign Language
ASL	American Sign Language
PIC	Peripheral Interface Controller
MATLAB	Matrix Laboratory
RGB	Red Green Blue
DIP	Digital Image Processing
ID	Identification Device
Bmp	Bit map image
SOM	Self Organizing Map
MM	Mathematical Morphology

LIST OF FIGURES

Fig.no	Title name	Page.no
1	BLOCK DIAGRAM OF HARDWARE SYSTEM	11
2	BLOCK DIAGRAM OF POWER SUPPLY	12
3	PIC 16F877A MICROCONTROLLER	13
4	CIRCUIT DIAGRAM	14
5	CIRCUIT DIAGRAM OF POWER SUPPLY	15
6	SCHEMATIC DIAGRAM OF FLEX SENSOR	17
7	VECTOR REPRESENTATION OF COLORS	23
8	FLOWCHART	25
9	DATABASE SAMPLES	28
10	RESULT FOR 'A'	40
11	RESULT FOR 'I'	40
12	RESULT FOR 'L'	41
13	RESULT FOR 'MAY I KNOW YOUR NAME?'	42
14	RESULT FOR 'PLEASE HELP ME'	42

ABSTRACT

Computer recognition of sign language is an important research problem for enabling communication with speaking impaired people. This project introduces an efficient and fast method for identification of the number of fingers opened in a gesture representing an alphabet and sentences of the Binary Sign Language and the American Sign Language. The system requires the hand to be perfectly aligned to the camera. The project uses image processing system to identify, especially English alphabetic sign language used by the speaking impaired people to communicate. The basic objective of this project is to develop a computer based intelligent system that will enable dumb people significantly to communicate with all other people using their natural hand gestures. The idea consisted of designing and building up an intelligent system using image processing concepts to take visual inputs of sign language's hand gestures and generate easily recognizable form of outputs. Hence the objective of this project is to develop an intelligent system which can act as a translator between the sign language and the spoken language dynamically and can make the communication between people with speaking impairment and normal people both effective and efficient. This project can be used for both American Sign Language and British Sign Language.

1. INTRODUCTION

Sign language is an expressive and natural way for communication between normal and dumb people (information majorly conveys through the hand gesture). Sign language uses both physical and non-physical communication. The physical gesture communication consist of hand gestures that convey respective meaning, the non physical is head movement, facial appearance, body orientation and position. Research in the sign language system has two well known approaches are Image processing and Data glove. The image processing technique uses the camera to capture the image/video. Another research approach is a sign language recognition system using a data glove. User need to wear glove consist of flex sensor and motion tracker. Data are directly obtained from each sensor depends upon finger flexures and computer analysis sensor data with static data to produce sentences. BSL and ASL is a visual language used by dumb people, which uses word level sign (gestures), non manual features. E.g. Finger spelling (letter by letter signing) to convey meaning. Although sign language is the main communication medium of the speaking impaired, in term of automatic recognition, finger spelling has the advantage of using limited number of finger signs, corresponding to the letters/sounds in the alphabets. Moreover as finger spelling is used in sign out of vocabulary words, the outcome of this project will provide modules that can be reused in a sign language to speech translator. The practical problems of recognizing finger spelling are very different from those of recognizing signs, in finger spelling there is only a fixed number of hand shapes that need to be recognized. The intension of the sign language translation system is to translate the normal sign language into speech and to make easy contact with the dumb people. In order to improve the life style of the dumb people the proposed system is developed.

1.1 HARDWARE APPROACH

The hardware approach makes use of flex sensor and WTV-SR voice recorder with PIC16F877A. The flex sensors respond to gesture. By using suitable circuit, response of the sensor is given to the microcontroller, based on the response microcontroller plays the recorded voice using the voice module. This system offers high reliability and fast response. This method is more precise on hand movement and different languages can be installed without altering the code in PIC microcontroller.

1.2 BLOCK DIAGRAM AND ITS DESCRIPTION

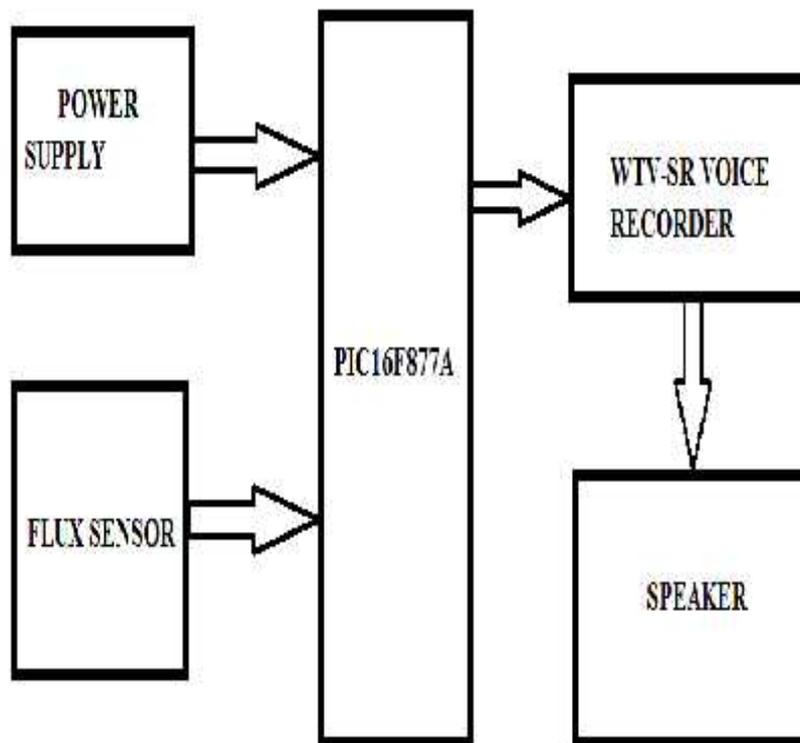


Fig.1 Block diagram of hardware system

1.2.1 POWER SUPPLY

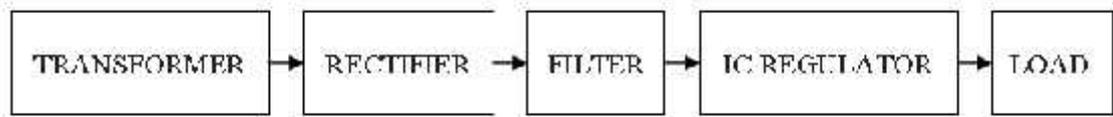


Fig.2 Block diagram of Power supply

The ac voltage, typically 220V rms, is connected to a transformer, which steps that ac voltage down to the level of the desired dc output. A diode rectifier then provides a full-wave rectified voltage that is initially filtered by a simple capacitor filter to produce a dc voltage. This resulting dc voltage usually has some ripple or ac voltage variation. A regulator circuit removes the ripples and also remains the same dc value even if the input dc voltage varies. This voltage regulation is usually obtained using one of the popular voltage regulator IC units.

1.2.2 FLEX SENSOR

In this system microcontroller receives data from the glove, it consists of 4.2 inch flex sensors. These sensors provide a corresponding signal of finger flexures and hand motion. Output data are directly obtained from data glove and each sensors produce different resistance value through the combination of resistance value. Then the respective voice plays according to the combination of resistance value.

1.3 CIRCUIT DIAGRAM AND ITS DESCRIPTION

SIGN LANGUAGE TO SPEECH TRANSLATION SYSTEM USING PIC MICROCONTROLLER

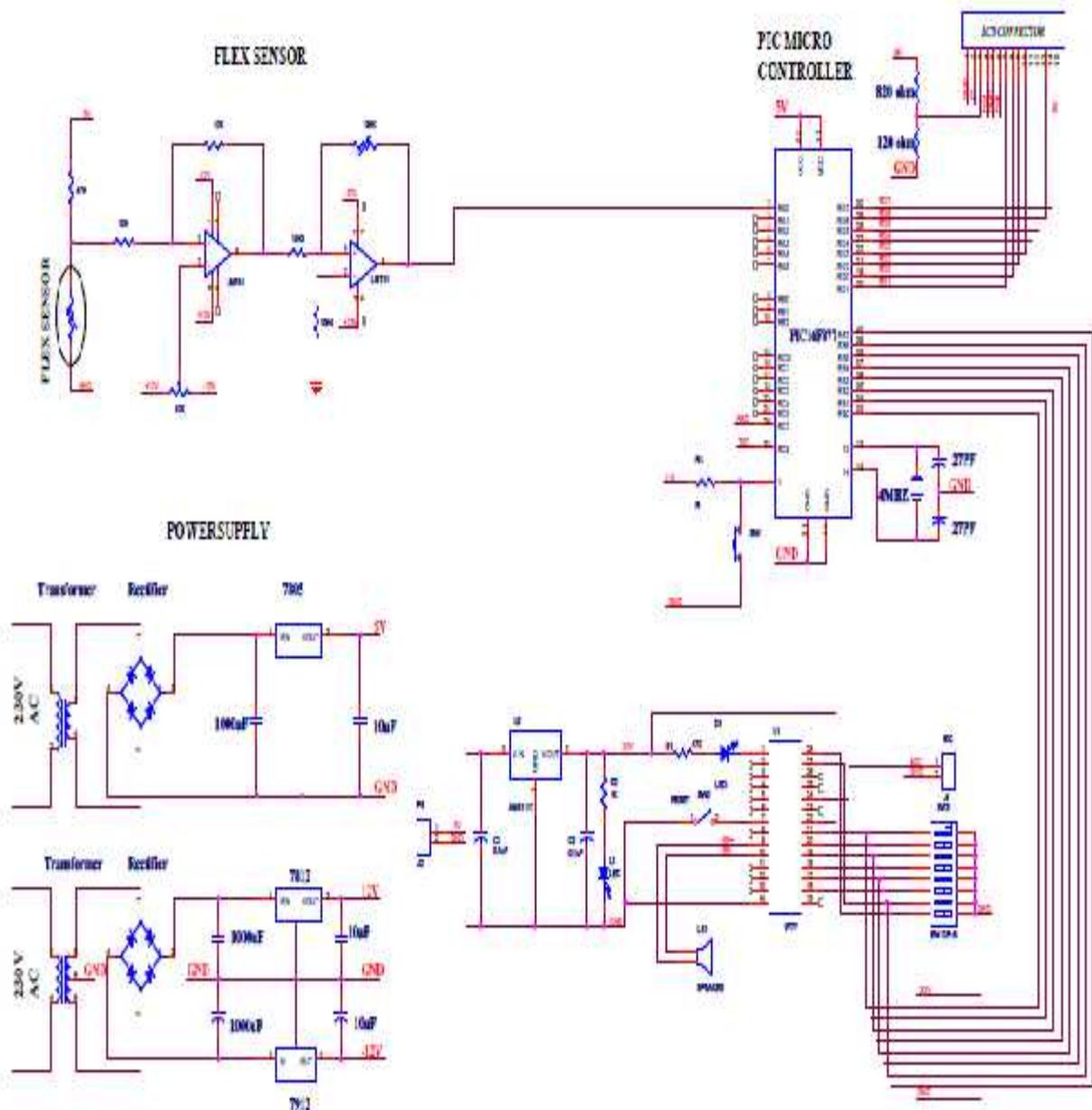


Fig.4 Circuit Diagram

1.3.1 POWER SUPPLY

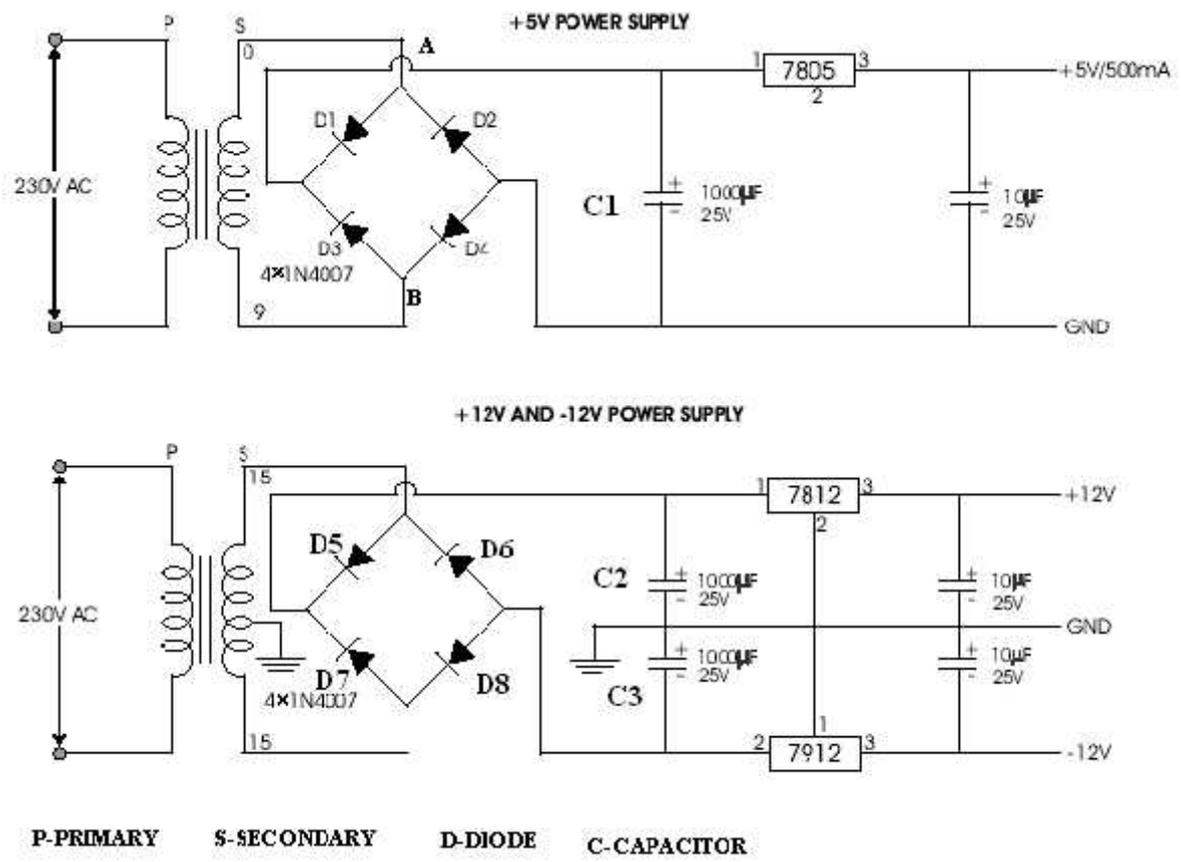


Fig.5 Circuit diagram of power supply

1.3.2 TRANSFORMER

The potential transformer will step down the power supply voltage (0-230V) to (0-9V and 15-0-15) level. This is a step down transformer. Then the secondary of the potential transformer will be connected to the rectifier.

1.3.3 BRIDGE RECTIFIER

The input to the circuit is applied to the diagonally opposite corners of the network, and the output is taken from the remaining two corners. There is a positive potential, at point A and a negative potential at point B. The positive potential at point A will forward bias D3 and reverse bias D4. The negative potential at point B will forward bias D1 and reverse D2. At this time D3 and

D1 are forward biased and will allow current flow to pass through them; D4 and D2 are reverse biased and will block current flow. The path for current flow is from point B through D1, up through Load, through D3, through the secondary of the transformer back to point B. One-half cycle later the polarity across the secondary of the transformer reverse, forward biasing D2 and D4 and reverse biasing D1 and D3. Current flow will now be from point A through D4, up through Load, through D2, through the secondary of transformer, and back to point A across D2 and D4. The current flow through load is always in the same direction. Since current flows through the load during both half cycles of the applied voltage, this bridge rectifier is a full-wave rectifier.

One advantage of a bridge rectifier over a conventional full-wave rectifier is that with a given transformer the bridge rectifier produces a voltage output that is nearly twice that of the conventional half-wave circuit. Since any two diodes are always conducting, total drop voltage is 1.4 volt.

1.3.4 FILTER

A Capacitor is added in parallel with the load resistor of a Rectifier to form a simple Filter Circuit. The output of the Rectifier will be transformed into a more stable DC Voltage. At first, the capacitor is charged to the peak value of the rectified Waveform. Beyond the peak, the capacitor is discharged through the load resistor until the time at which the rectified voltage exceeds the capacitor voltage. Then the capacitor is charged again and the process repeats itself.

1.3.5 IC VOLTAGE REGULATORS

A fixed three-terminal voltage regulator has an unregulated dc input voltage , applied to one input terminal, a regulated dc output voltage, from a second terminal, with the third terminal connected to ground. A regulated power supply circuit using the 78xx IC series is used here. These regulators can deliver

current around 1A to 1.5A at a fix voltage levels. In this power supply circuit 5, 12 and -12Volt output is obtained.

1.3.6 FLUX SENSOR

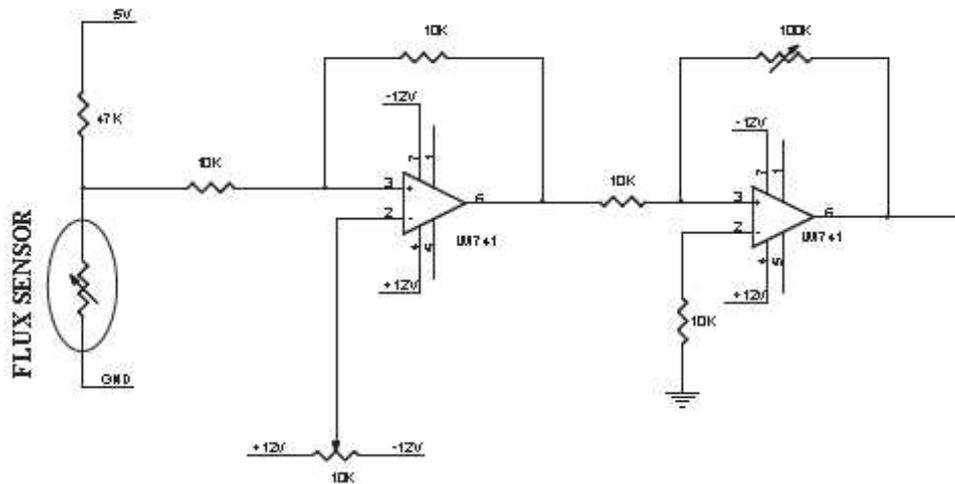


Fig.6 Schematic Diagram of flex sensor

$$V_{out} = V_{in} \frac{R_2}{(R_1 + R_2)}$$

If the R1 and R2 value is equal means the output is half of the Vcc supply. In this circuit output is a variable one. So the output is depending upon the R2 resistance value. Resistance will be varied depending upon the fluctuation level. If fluctuation level varies then the resistance value also varies. If resistance value is increased, output is also increased. The resistance value and output is a directly proportional one.

1.3.7 AMPLIFIERS

The output is given to the inverting amplifiers inverting pin. So it amplifies the electrical signal and also inverts the signal to 180 degree. Then the output is given to another one inverting terminal. So again the electrical signal is amplified and inverts 180 degree. Now the original signal is

obtained($180+180=360$). Then the final voltage is given to ADC for converting the analog signal to digital signal. Then the corresponding digital signal is taken to process in microcontroller. The ADC value will increase if the fluctuation increased.

1.3.8 PIC 16F877 MICROCONTROLLER

PIC microcontroller contain 10-bit inbuilt ADC and use to receive the analog value from the sensor. An ADC converts analog to digital value and store the value in the buffer. Technology that is used in PIC 16F877 is flash technology, so that data is retained even when the power is switched off. Easy Programming and Erasing are other features of PIC 16F877.

1.3.9 WTV-SR VOICE RECORDER

The WTV Voice module Board is designed to work in parallel port mode. In the Parallel Port Mode, Module can record 32 Group of voice. To record/playback the voice 8 pin DIP switch is used. Each pin in DIP switch has a different function which has been listed below.

- PIN 01- for Record.
- PIN 02-for Play.
- PIN 03-for Erase
- PIN 04, PIN 05, PIN 06, PIN 07 and PIN 08- for selecting address.

1.4 WORKING PRINCIPLE

The power supply unit consists of a step down transformer, rectifier, filter, IC regulator. The transformer steps down the voltage from 230v to 5v ac and supplies it to the rest of the circuit. The Bridge rectifier circuit converts the AC voltage into DC voltage and the IC regulator circuit removes the ripples.

The sensing units consists of the flux sensor whose resistance value changes based on the hand gestures. The processing unit consists of the PIC 16F877A microcontroller. PIC microcontroller contain 10-bit inbuilt ADC which receives the analog values from the sensor. An ADC converts analog to digital value and store the value in the buffer. Then Controller compares the static data and digital value for processing to determine the gesture. According to the finger movements microcontroller plays the voice (speech). Playback capacity of WTV-SR is 40 secs. Here a crystal oscillator of 4MHZ is used to provide clock pulses. Finally the voice signal corresponding to the gesture is played.

1.5 DISADVANTAGES OF HARDWARE APPROACH

- In hardware approach, the system design is complex.
- Since the hardware makes use of flex sensors, data gloves, contact sensors and accelerometers, it is difficult for it to be carried around.
- Hardware approach is not economical.
- If any error occurs in the hardware system, it might result in the entire replacement of the sensors. This will lead to additional cost.
- The hardware system is not reliable.
- Hardware system is not efficient since the resistance values of the flex sensors cannot be specifically determined.
- Hardware system is less durable.
- The computational time is high.

2. SOFTWARE APPROACH

The main idea for opting software approach is to make the system easily available and to reduce the design complexity. This approach increases the storage capacity and also reduces the computational time. Since it is in the form of program coding, the errors can be easily identified and rectified. The durability of this system is longer than the hardware system. Thus we choose to carry out the approach in image processing.

The software approach makes use of web camera and MATLAB for processing the sign shown by the user. First the background is captured by the web camera. It should be a plain background. After the background is captured, it gets saved in the database. When the sign is shown by the user, the image gets compared with the samples already stored in the database. The corresponding alphabets or sentences are displayed and the audio is played.

2.1 DIGITAL IMAGE PROCESSING

The identification of objects in an image would probably start with image processing techniques such as noise removal, followed by (low-level) feature extraction to locate lines, regions and possibly areas with certain textures. The technique is to interpret collections of these shapes as single objects, e.g. cars on a road, boxes on a conveyor belt or cancerous cells on a microscope slide. One reason this is a problem is that an object can appear very different when viewed from different angles or under different lighting. Another problem is deciding what features belong to what object and which are background or shadows etc. The human visual system performs these tasks mostly unconsciously but a computer requires skilful programming and lots of processing power to approach human performance. For manipulating data in the form of an image through several possible techniques, an image is usually interpreted as a two-dimensional array of brightness values, and is most familiarly represented by

such patterns as those of a photographic print, slide, television screen, or movie screen. An image can be processed optically or digitally with a computer.

2.2 CLASSIFICATION OF IMAGES:

There are 3 types of images used in Digital Image Processing. They are

1. Binary Image
2. Gray Scale Image
3. Color Image

2.2.1 BINARY IMAGE:

A binary image is a digital image that has only two possible values for each pixel. Typically the two colors used for a binary image are black and white though any two colors can be used. The color used for the object(s) in the image is the foreground color while the rest of the image is the background color.

Binary images are also called bi-level or two-level. This means that each pixel is stored as a single bit (0 or 1). This name black and white, monochrome or monochromatic are often used for this concept, but may also designate any images that have only one sample per pixel, such as grayscale images

Binary images often arise in digital image processing as masks or as the result of certain operations such as segmentation, thresholding, and dithering. Some input/output devices, such as laser printers, fax machines, and bi-level computer displays, can only handle bi-level images

2.2.2 GRAY SCALE IMAGE:

A grayscale Image is digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information. Images of this sort, also known as black-and-white, are composed exclusively of shades of gray(0-255), varying from black(0) at the weakest intensity to white(255) at the strongest.

Grayscale images are distinct from one-bit black-and-white images, which in the context of computer imaging are images with only the two colors, black, and white (also called bi-level or binary images). Grayscale images have many shades of gray in between. Grayscale images are also called monochromatic, denoting the absence of any chromatic variation.

Grayscale images are often the result of measuring the intensity of light at each pixel in a single band of the electromagnetic spectrum (e.g. infrared, visible light, ultraviolet, etc.), and in such cases they are monochromatic proper when only a given frequency is captured. But also they can be synthesized from a full color image.

2.2.3 COLOR IMAGE:

A (digital) color image is a digital image that includes color information for each pixel. Each pixel has a particular value which determines its appearing color. This value is qualified by three numbers giving the decomposition of the color in the three primary colors Red, Green and Blue. Any color visible to human eye can be represented this way. The decomposition of a color in the three primary colors is quantified by a number between 0 and 255. For example, white will be coded as $R = 255, G = 255, B = 255$; black will be known as $(R,G,B) = (0,0,0)$; and say, bright pink will be : $(255,0,255)$.

In other words, an image is an enormous two-dimensional array of color values, pixels, each of them coded on 3 bytes, representing the three primary colors. This allows the image to contain a total of $256 \times 256 \times 256 = 16.8$ million different colors. This technique is also known as RGB encoding, and is specifically adapted to human vision.

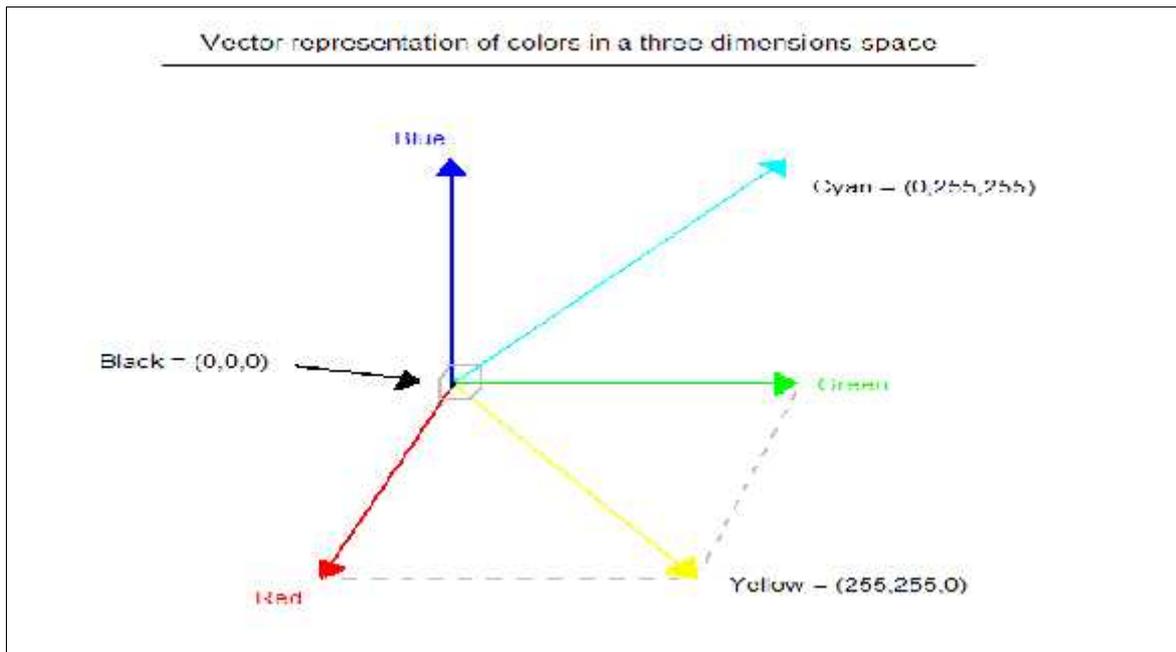


Fig.7 Vector representation of colors

From the above figure, colors are coded on three bytes representing their decomposition on the three primary colors. It sounds obvious to a mathematician to immediately interpret colors as vectors in a three dimension space where each axis stands for one of the primary colors.

2.3 MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Math and computation
- Algorithm development
- Modelling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB uses software developed by the LAPACK and ARPACK projects, which together represent the state-of-the-art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow us to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

2.4 FLOWCHART AND ITS DESCRIPTION:

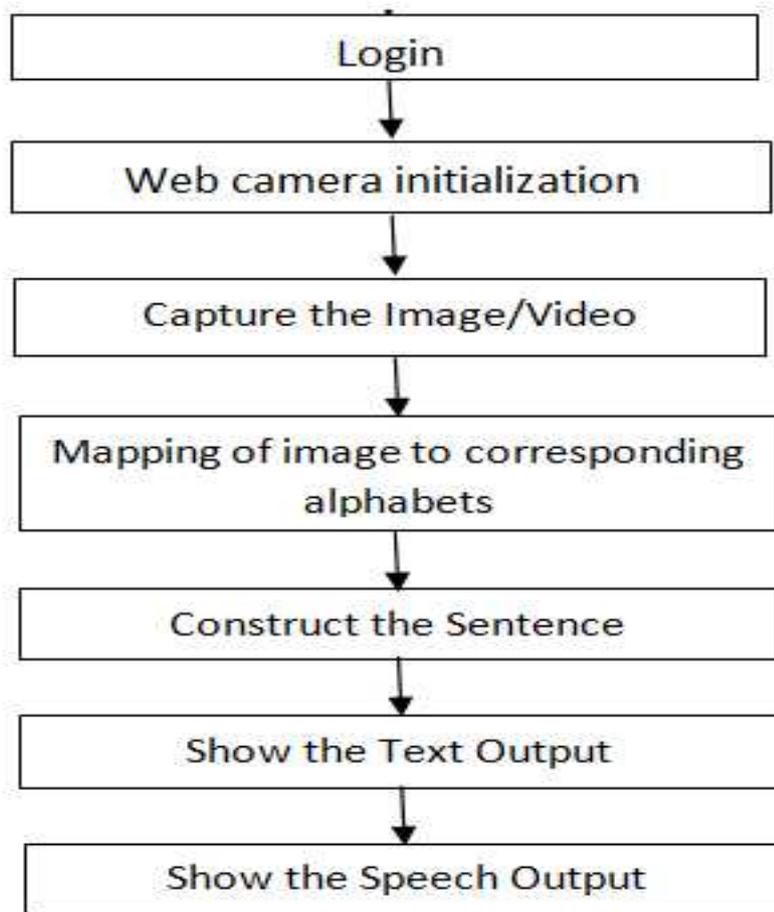


Fig.8 Flowchart

- Switch on the computer and open the MATLAB software.
- Initialise the web camera by specifying the web camera properties.
- The background image is first captured.
- After few counts the sign shown by the user is captured by the web camera.
- The predefined database samples are compared with the sign shown by the user.
- The required alphabets, words and sentences are stored.
- During the run time of the program, based on the image matching, the output is displayed as text and played as audio.

2.5 ALGORITHM

- Specify the adaptor name, device ID and format.
- Set the input video.
- Read the mask image with extension .bmp
- Convert the mask image to binary image.
- Set the boundaries and plot using vector values.
- Display the boundary.
- To start the video, the count is set to 1.
- The video is started and the background is captured when the count is 9.
- The background is separated to red, green and blue.
- The captured background is saved as "bg.bmp".
- The count is incremented to capture the sign.
- The video is stopped when the count is 20.
- The captured sign image is differentiated to red, green and blue.
- It is saved as "final.bmp".
- In segmentation process, the captured sign image is converted to grey scale image.
- In grey scale, the background is converted to black and hand sign is converted to white.
- Median filter is implemented for noise reduction.
- The area of the captured sign image is compared to that of the database.
- There are 5 to 10 samples for each alphabet.
- The database image which coincides with the captured sign image is taken into account.
- The corresponding alphabet is displayed.

2.6 WORKING PRINCIPLE

- It is important to carefully choose the direction in which the camera points to permit an easy choice of background.

- The realistic options are to point the camera towards a wall or a white background.
- The plain background is detected and then captured after a specific count.
- Now the sign is shown within the boundary.
- The image gets saved and the background is subtracted to extract the sign.
- This is compared to the already stored sample images.
- The corresponding text is displayed and the speech is played.

2.7 IMAGE DATA BASE

The starting point of the project was the creation of a data base with all the images that would be used for training and testing. The image data can have different formats. Images can be either hand drawn, digitized photographs or a 3D dimensional hand. Photographs were used, as they are the most realistic approach. Images came from two main sources: Myanmar Alphabet Language databases on the Internet and photographs that are taken with a digital camera. This meant that they have different sizes, different resolutions and sometimes almost completely different angles of shooting. Two operations were carried out in all of the images. They were converted to grayscale and the background was made uniform. The internet databases already had uniform backgrounds but the ones is taken with the digital camera had to be processed in Adobe Photoshop. Drawn images can still simulate translational variances with the help of an editing program (e.g. Adobe Photoshop). The database itself was constantly changing throughout the completion of the project as it would decide the robustness of the algorithm. Therefore, it had to be done in such way that different situations could be tested and thresholds above which the algorithm didn't classify correct would be decided. The construction of such a database is clearly dependent on the application. If the application is a crane controller for example operated by the same person for long periods the algorithm doesn't

have to be robust on different person's images. In this case noise and motion blur should be tolerable. For most of the gestures the training set originates from a single gesture. Those enhanced in Adobe Photoshop using various filters. The reason for this is to get the algorithm to be very robust for images of the same database.

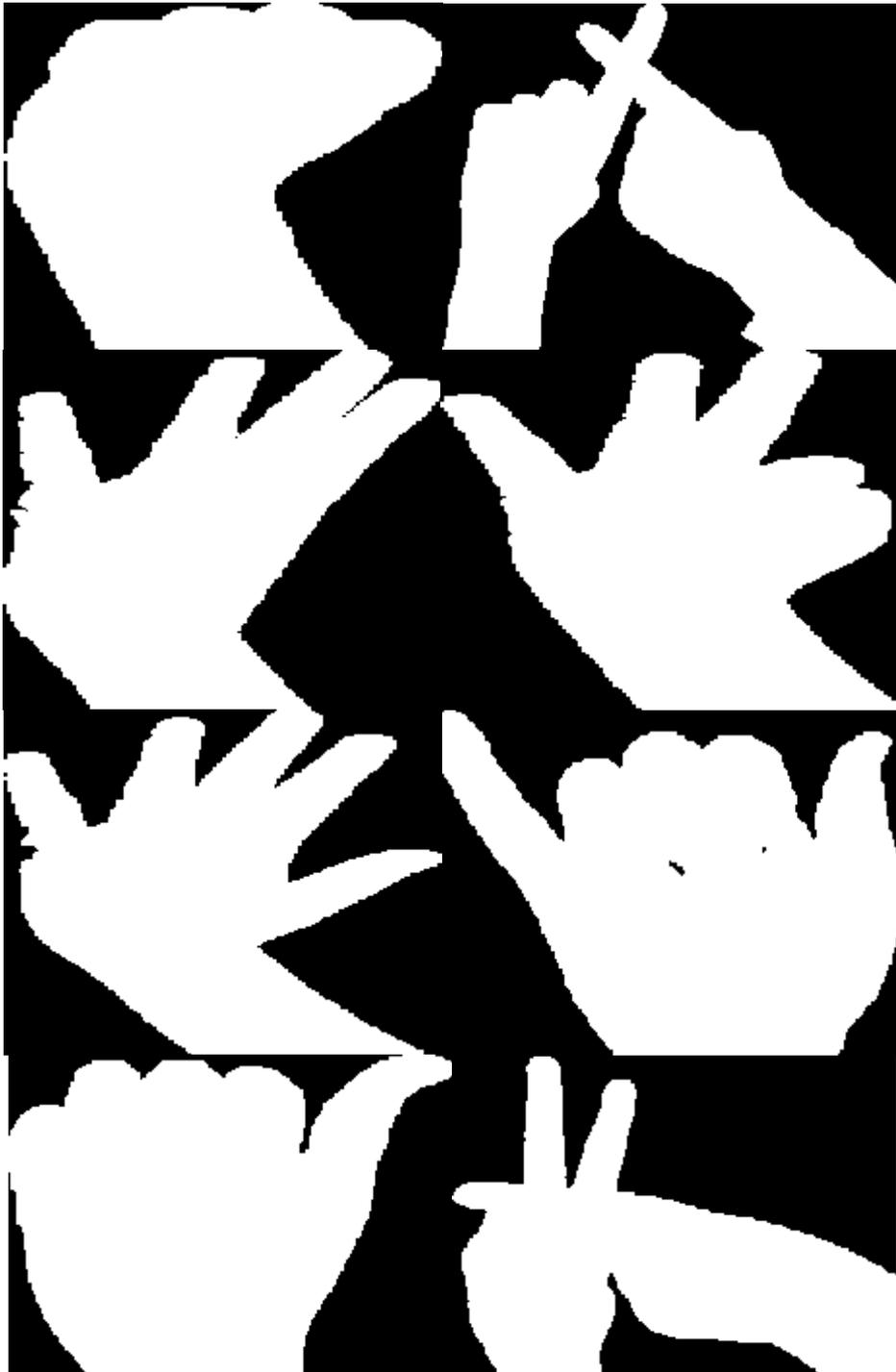


Fig.9 Database samples

2.8 LARGE OBJECT TRACKING

In some interactive applications, the computer needs to track the position or orientation of a hand that is prominent in the image. Relevant applications might be computer games, or interactive machine control. In such cases, a description of the overall properties of the image may be adequate. If the hand is on a uniform background, Large Object Tracking method can distinguish hand positions and simple pointing gestures.

2.9 SEGMENTATION

2.9.1 REGION GROWING APPROACH

Region growing technique segments image pixels that belong to an object into regions. Segmentation is performed based on some predefined criteria. Two pixels can be grouped together if they have the same intensity characteristics or if they are close to each other. It is assumed that pixels that are closed to each other and have similar intensity values are likely to belong to the same object. The simplest form of the segmentation can be achieved through thresholding and component labelling. Another method is to find region boundaries using edge detection. Segmentation process, then, uses region boundary information to extract the regions. The main disadvantage of region growing approach is that it often requires a seed point as the starting point of the segmentation process. This requires user interaction. Due to the variations in image intensities and noise, region growing can result in holes and over segmentation. Thus, it sometimes requires post-processing of the segmentation result.

A hierarchical set of image segmentations is a set of several image segmentations of the same image at different levels of detail in which the segmentations at coarser levels of detail can be produced from simple merges of regions at finer levels of detail. A unique feature of hierarchical segmentation is that the segment or region boundaries are maintained at the full image spatial

resolution for all segmentations. In a hierarchical segmentation, an object of interest may be represented by multiple image segments in finer levels of detail in the segmentation hierarchy, and may be merged into a surrounding region at coarser levels of detail in the segmentation hierarchy. If the segmentation hierarchy has sufficient resolution, the object of interest will be represented as a single region segment at some intermediate level of segmentation detail.

A goal of the subject analysis of the segmentation hierarchy is to identify the hierarchical level at which the object of interest is represented by a single region segment. The object may then be identified through its spectral and spatial characteristics. Additional clues for object identification may be obtained from the behavior of the image segmentations at the hierarchical segmentation level above and below the level at which the object of interest is represented by a single region.

The simplest method of image segmentation is called the thresholding method. This method is based on a clip-level (or a threshold value) to turn a gray-scale image into a binary image. The key of this method is to select the threshold value (or values when multiple-levels are selected). Several popular methods are used in industry including the maximum entropy method, Otsu's method (maximum variance), and k-means clustering. Recently, methods have been developed for thresholding computed tomography (CT) images. The key idea is that, unlike Otsu's method, the thresholds are derived from the radiographs instead of the (reconstructed) image.

2.9.2 DRAWBACKS

- Object classification accuracy is less
- Sensitive to noise
- It is not suitable for all lighting conditions of images

2.10 BACKGROUND MODELLING

Although there are many algorithms for background subtraction, they all follow a general pattern of processing which includes pre-processing, background subtraction, post processing.

2.10.1 PREPROCESSING

Firstly, video frames captured from a camera are input to the background subtractor. Preprocessing stages are used for filtration and to change the raw input video to a processable format .

2.10.2 BACKGROUND SUBTRACTION

The main problem of the background subtraction approach to moving object detection is its extreme sensitivity to dynamic scene changes due to lighting and extraneous events. Although these are usually detected, they leave behind “holes” where the newly exposed background imagery differs from the known background model (ghosts). While the background model eventually adapts to these “holes,” they generate false alarms for a short period of time. Therefore, it is highly desirable to construct an approach to motion detection based on a background model that automatically adapts to changes in a self-organizing manner and without *a priori* knowledge.

We propose to adopt a biologically inspired problem-solving method based on visual attention mechanisms. The aim is to obtain the objects that keep the user attention in accordance with a set of pre-defined features, including gray level, motion and shape features. Our approach defines a method for the generation of an active attention focus to monitor dynamic scenes for surveillance purposes. The idea is to build the background model by learning in a self-organizing manner many background variations, i.e., background motion cycles, seen as trajectories of pixels in time. Based on the learnt background

model through a map of motion and stationary patterns, our algorithm can detect motion and selectively update the background model. Specifically, a novel neural network mapping method is proposed to use a whole trajectory incrementally in time fed as an input to the network. This makes the network structure much simpler and the learning process much more efficient.

The adopted artificial neural network is organized as a 2-D flat grid of neurons (or nodes) and, similarly to self organizing maps (SOMs) or Kohonen networks, allows to produce representations of training samples with lower dimensionality, at the same time preserving topological neighbourhood relations of the input patterns (nearby outputs correspond to nearby input patterns). Each node computes a function of the weighted linear combination of incoming inputs, where weights resemble the neural network learning. Doing so, each node could be represented by a weight vector, obtained collecting the weights related to incoming links. In the following, the set of weight vectors will be called a *model*. An incoming pattern is mapped to the node whose model is “most similar” (according to a pre-defined metric) to the pattern, and weight vectors in a neighbourhood of such node are updated. Therefore, the network behaves as a competitive neural network that implements a winner take-all function with an associated mechanism, that modifies the local synaptic plasticity of the neurons, allowing learning to be restricted spatially to the local neighbourhood of the most active neurons. For each color pixel, we consider a neuronal map consisting of $n \times n$ weight vectors. Each incoming sample is mapped to the weight vector that is closest according to a suitable distance measure, and the weight vectors in its neighbourhood are updated. The whole set of weight vectors acts as a background model that is used for background subtraction in order to identify moving pixels.

2.10.3 POST PROCESSING

Background modelling then uses the observed video frame to calculate and to update the background model that is representative of the scene without any objects of interest. Foreground detection is where the pixels that show a significant difference to those in the background model are flagged as foreground. Data validation is used to examine the found objects of interest and to eliminate any false matches. A foreground mask can then be output in which pixels are assigned as foreground or background. The foreground detection stage can be described as a binary classification problem whereby each pixel in an image is assigned a label to the class of foreground or background. Formally, for every pixel p in image I , a label p_l is assigned where $I \in \{0, 1\}$ where $0 =$ background and $1 =$ foreground. After this mask is obtained, background pixels are usually set to white or black to allow focus on the foreground object. Many simple decision rules to classify each pixel have been suggested, each of which can be carried out in any one of a number of colour spaces. Many background subtraction algorithms reduce down to the simple subtraction of the pixel in the expected background image from the pixel in the observed image and any significant change indicates that an object of interest has been identified.

2.11 SHAPE RECOGNITION

Most applications, such as recognizing particular static hand signal, require the shape of the input object than image moments provide. If the hand signals fell in a predetermined set, and the camera views a close-up of the hand, combined with a simple method to analyze hand signals called World Academy of Science, Engineering and Technology 50 2009 orientation histograms. These applications involve two phases; training and running. In the training phase, the user shows the system one or more examples of a specific hand shape. The computer forms and stores the corresponding orientation

histograms. In the run phase, the computer compares the orientation histogram of the current image with each of the stored templates and selects the category of the closest match, or interpolates between templates, as appropriate. This method should be robust against small differences in the size of the hand but probably would be sensitive to changes in hand orientation.

2.11.1 DRAWBACKS

- Detection of object is not optimal for all lighting conditions of an image.
- Lack of representing shape of object regions.
- Low recognition accuracy.

2.12 ADVANTAGES OF SOFTWARE APPROACH

- In software approach, the system complexity is minimised.
- Software approach makes use of web camera in laptop. So it is portable when compared to hardware approach.
- Software approach is economical.
- In software system, it is easier to identify the error and rectify the mistakes done in the coding.
- Software system is reliable.
- Software approach is accurate and efficient.
- Software system has high durability.

3. METHODOLOGIES

- Preprocessing – Smoothing process
- Morphological filtering
- Connected Component Analysis
- Correlation Coefficient

3.1 GAUSSIAN SMOOTHING PROCESS

A Gaussian smoothing is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise and reduce detail. Gaussian smoothing is also used as a pre-processing stage in computer vision algorithms in order to enhance image structures at different scales.

Mathematically, applying a Gaussian blur to an image is the same as convolving the image with a Gaussian function. By contrast, convolving by a circle (i.e., a circular box blur) would more accurately reproduce the bokeh effect. Since the Fourier transform of a Gaussian is another Gaussian, applying a Gaussian blur has the effect of reducing the image's high-frequency components; a Gaussian blur is thus a low pass filter.

The idea of Gaussian smoothing is to use this 2-D distribution as a 'point-spread' function, and this is achieved by convolution. Since the image is stored as a collection of discrete pixels we need to produce a discrete approximation to the Gaussian function before we can perform the convolution. In theory, the Gaussian distribution is non-zero everywhere, which would require an infinitely large convolution kernel, but in practice it is effectively zero more than about three standard deviations from the mean, and so we can truncate the kernel at this point.

$$\text{Gauss Coeff} = (1/\sqrt{2*\pi*\text{sig}^2})(\exp (-x^2+y^2/2*\text{sig}^2))$$

Where, x , y , σ - input coordinates corresponds to the target and standard Deviation.

3.2 MATHEMATICAL MORPHOLOGY

Mathematical morphology (MM) is a theory and technique for the analysis and processing of geometrical structures, based on set theory, lattice theory, topology, and random functions. MM is most commonly applied to digital images, but it can be employed as well on graphs, surface meshes, solids, and many other spatial structures.

Topological and geometrical continuous-space concepts such as size, shape, convexity, connectivity, and geodesic distance, can be characterized by MM on both continuous and discrete spaces. MM is also the foundation of morphological image processing, which consists of a set of operators that transform images according to the above characterizations.

MM was originally developed for binary images, and was later extended to grayscale functions and images. The subsequent generalization to complete lattices is widely accepted today as MM's theoretical foundation.

3.2.1 BINARY MORPHOLOGY

In binary morphology, an image is viewed as a subset of an Euclidean space \mathbb{R}^d or the integer grid \mathbb{Z}^d , for some dimension d . The basic idea in binary morphology is to probe an image with a simple, pre-defined shape, drawing conclusions on how this shape fits or misses the shapes in the image. This simple "probe" is called structuring element, and is itself a binary image (i.e., a subset of the space or grid).

3.3 CONNECTED COMPONENT ANALYSIS

The output of the change detection module is the binary image that contains only two labels, i.e., '0' and '255', representing as 'background' and 'foreground' pixels respectively, with some noise. The goal of the connected component analysis is to detect the large sized connected foreground region or object. This is one of the important operations in motion detection. The pixels that are collectively connected can be clustered into changing or moving objects by analyzing their connectivity.

In binary image analysis, the object is extracted using the connected component labelling operation, which consist of assigning a unique label to each maximally connected Foreground region of pixels. One of the important labelling approaches is "classical sequential labelling algorithm" . It is based on two raster scan of binary image. The first scan performs the temporary labelling to each foreground region pixels by checking their connectivity of the scanned image. When a foreground pixel with two or more than two foreground neighbouring pixels carrying the same label is found, the labels associated with those pixels are registered as being equivalent. That means these regions are from the same object. The handling of equivalent labels and merging thereafter is the most complex task.

The first scan gives temporary labels to the foreground pixels according to their connectivity. The connectivity check can be done with the help of either a 4-connectivity or 8-connectivity approach. 8-connectivity approach is used. Here, the idea is to label the whole blob at a time to avoid the label redundancies.

The labelling operation scans the image moving along the row until it comes to the point P , for which $S = \{255\}$. When this is true, it checks the four neighbours of which Based on that information, the labelling of P occurs as follows,

- If all four neighbours are '0' assign a new label to P, and increment the label,

Else

If only one neighbour has $S = \{255\}$ assigns its label to P

Else (i.e., more than one of the neighbours has $S = \{255\}$) Assign one of the labels to P.

Here, note that the relation between the pixels that are expressed through a "label value" in the labelled image depends on the value of the label. That means the two pixels from Background, labelled as l_B is not necessarily to be connected, but the two pixels labelled l_P from the foreground region are to be connected.

3.3.1 LOCAL REGION DESCRIPTORS

The Labelled objects within a sign are applied to measure its characteristics which are useful to recognize a sign with stored templates. The following features are extracted, Area, Orientation, Height, width, Eccentricity, perimeter and Equivalent diameter

3.3.2 TEMPLATE MATCHING

It is a technique in digital image processing for finding small parts of an image which match a template image. A sliding window over other image sequences is used to indicate the possible presence of the reference target. A regional feature matching operator is applied to find the similarity between the target model and the pixels within the window. The labelled component from segmentation module will be applied to extract the region features to describe its characteristics. Here correlation coefficient will be used to measure the similarity between two different objects for target detection and tracking.

3.4 CORRELATION COEFFICIENT

It is used to find the similarity between two different objects with their region features. It will be described by,

$$\text{Cor_coef} = \frac{[\text{sum}(\text{sum}(\mathbf{u1}.*\mathbf{u2}))]}{[\text{sqrt}(\text{sum}(\text{sum}(\mathbf{u1}.*\mathbf{u1}))*\text{sum}(\text{sum}(\mathbf{u2}.*\mathbf{u2})))];}$$

Where, $u1 = F1 - \text{mean of } F1$, $u2 = F2 - \text{mean of } F2$

$F1 - \text{Feature set1}$ and $F2 - \text{Features set2}$

3.4.1 CROSS CORRELATION

Cross correlation is a standard method of estimating the degree to which two series are correlated. Consider two series $x(i)$ and $y(i)$ where $i=0,1,2\dots N-1$. The cross correlation r at delay d is defined as

$$r = \frac{\sum_1 [(x(i) - m_x) * (y(i-d) - m_y)]}{\sqrt{\sum_1 (x(i) - m_x)^2} \sqrt{\sum_1 (y(i-d) - m_y)^2}}$$

Where m_x and m_y are the means of the corresponding series. If the above is computed for all delays $d=0,1,2,\dots N-1$ then it results in a cross correlation series of twice the length as the original series.

$$r(d) = \frac{\sum_1 [(x(i) - m_x) * (y(i-d) - m_y)]}{\sqrt{\sum_1 (x(i) - m_x)^2} \sqrt{\sum_1 (y(i-d) - m_y)^2}}$$

4. RESULTS

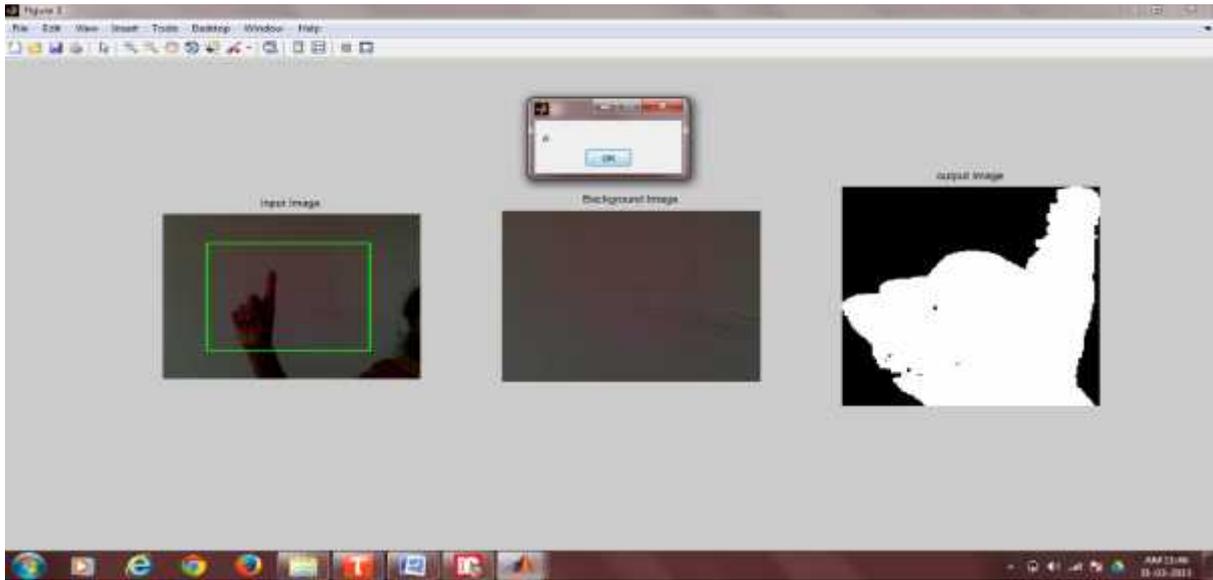


Fig.10 Result for 'A'

- The input image is given as the sign 'A' shown by the user within the green boundary.
- The background image is captured and displayed.
- The output image is the sign shown in gray scale.
- The alphabet 'A' is displayed and speech output is played.

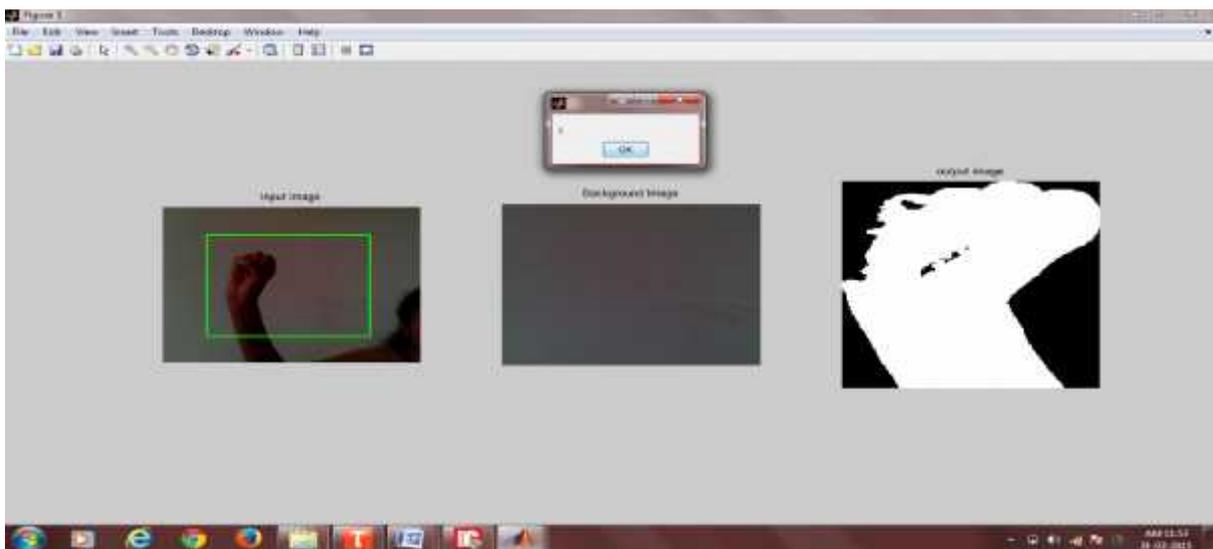


Fig.11 Result for 'T'

- The input image is given as the sign 'I' shown by the user within the green boundary.
- The background image is captured and displayed.
- The output image is the sign shown in gray scale.
- The alphabet 'I' is displayed and speech output is played.

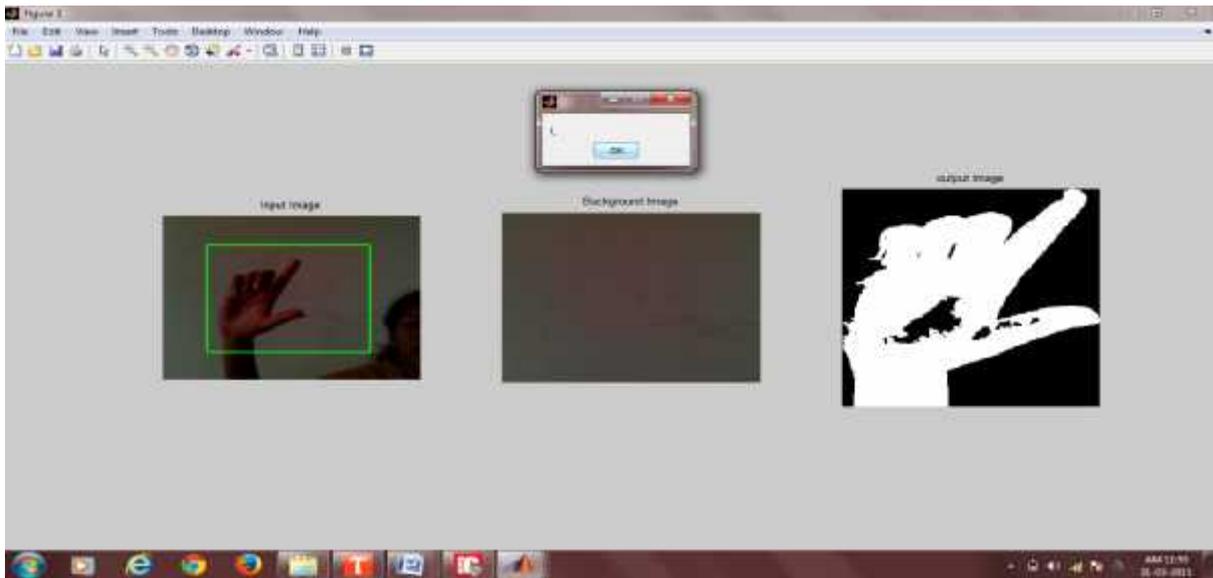


Fig.12 Result for 'L'

- The input image is given as the sign 'L' shown by the user within the green boundary.
- The background image is captured and displayed.
- The output image is the sign shown in gray scale.
- The alphabet 'L' is displayed and speech output is played.

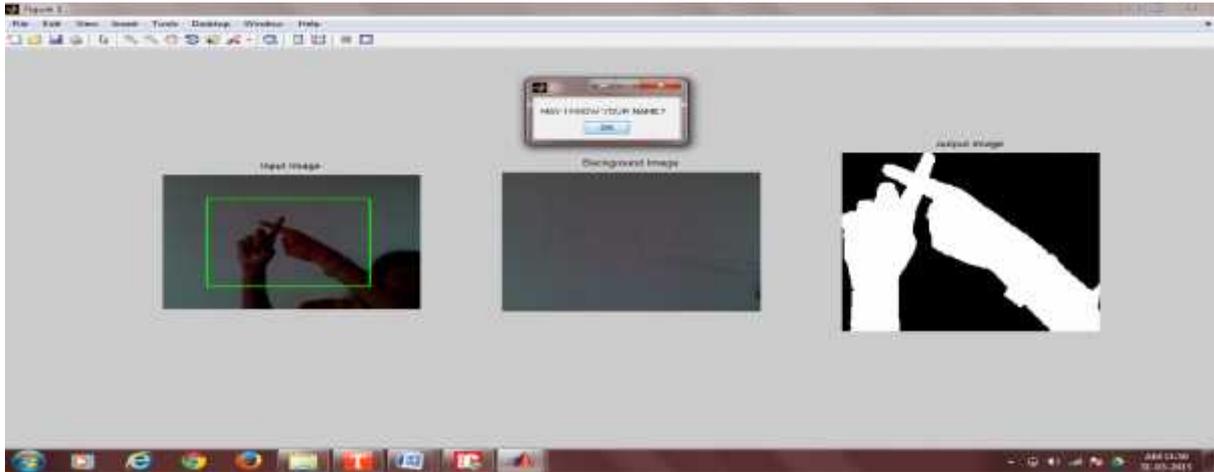


Fig.13 Result for ‘MAY I KNOW YOUR NAME?’

- The input image is given as the sign for ‘MAY I KNOW YOUR NAME?’ shown by the user within the green boundary.
- The background image is captured and displayed.
- The output image is the sign shown in gray scale.
- The question ‘MAY I KNOW YOUR NAME?’ is displayed and speech output is played.

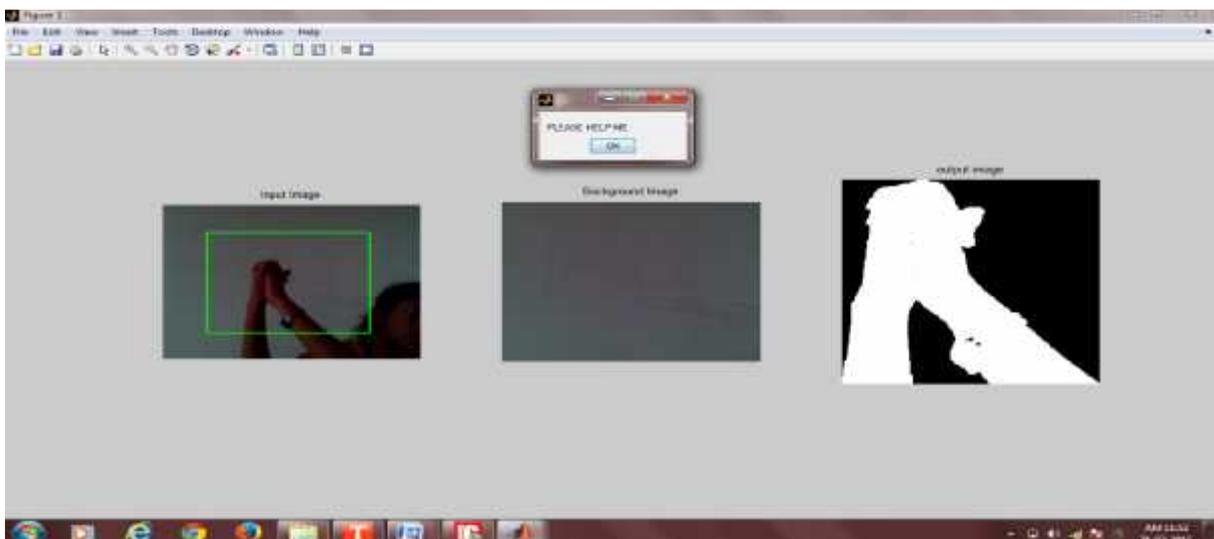


Fig.14 Result for ‘PLEASE HELP ME’

- The input image is given as the sign for 'PLEASE HELP ME' shown by the user within the green boundary.
- The background image is captured and displayed.
- The output image is the sign shown in gray scale.
- The sentence 'PLEASE HELP ME' is displayed and speech output is played.

5. CONCLUSION AND FUTURE WORKS

The proposed method translates sign language to speech automatically and satisfies them by conveying thoughts on their own. The system overcomes the real time difficulties of dumb people and improves their lifestyle. In this proposed work, we present a simple model of static gestures of British Sign Language and American Sign Language alphabet recognition system, which does not require any special markers or gloves and can operate in real-time on a commodity PC with low cost cameras. Compared with existing system it's possible to carry to any places. In future there are possibilities of implementing in domestic applications, wearable computers, web conference and video games.

6. APPENDIX

PROGRAM CODING

```
clc;
closeall;
clearall;
imaqhwinfo
info = imaqhwinfo('winvideo',1);
celldisp(info.SupportedFormats)
vid=videoinput('winvideo',1,'YUY2_640x480');
set(vid,'ReturnedColorSpace','rgb');
BW = imread('mask3.jpg');
BW=im2bw(BW);
[B,L,N,A] = bwboundaries(BW);
subplot(1,3,1);
imshow(BW);
title('Input Image');
holdon;

for k=1:length(B),
for l=find(A(:,k))'
boundary = B{l};
subplot(1,3,1);
saveboundaryboundary
plot(boundary(:,2), boundary(:,1), 'g','LineWidth',2);
end
end
```

```

loadboundary
aa=1;
r=81:402;
c=107:519;

while(1)
im=getsnapshot(vid);
subplot(1,3,1);
imshow(im);
holdon
ifaa == 4
red = im(:,:,1);
    Green = im(:,:,2);
    Blue = im(:,:,3);
Out(:,:,1)=red(r,c);
Out(:,:,2)=Green(r,c);
Out(:,:,3)=Blue(r,c);
    Out=uint8(Out);
subplot(1,3,2);
imshow(Out);
title('Background Image');
imwrite(Out,'bg.bmp');
end

subplot(1,3,1);
plot(boundary(:,2), boundary(:,1), 'g','LineWidth',2);
aa=aa+1;
disp(aa);
ifaa == 10

```

```
break
```

```
end
```

```
end
```

```
stop(vid),delete(vid),clear vid;
```

```
red = im(:,:,1);
```

```
Green = im(:,:,2);
```

```
Blue = im(:,:,3);
```

```
Out(:,:,1)=red(r,c);
```

```
Out(:,:,2)=Green(r,c);
```

```
Out(:,:,3)=Blue(r,c);
```

```
Out=uint8(Out);
```

```
imwrite(Out,'final.bmp');
```

```
a=imread('bg.bmp');
```

```
[C1,c1]=segment(a,Out);
```

```
subplot(1,3,3);
```

```
imshow(C1,[]);
```

```
title('output image');
```

```
imwrite(C1,'z3.bmp');
```

```
str='.bmp';
```

```
fori=1:139
```

```
    a=strcat(num2str(i),str);
```

```
    b=imread(a);
```

```
    re1=corr2(b,C1);
```

```
fresultNames_r(i) = {a};
```

```
result1(i)=re1;
```

```
end
```

```
[re ma] = max(result1);  
if ma > 0 && ma <=10  
msgbox('A');  
    [y, Fs]=wavread('A.wav');  
sound(y, Fs);  
elseif ma > 10 && ma <=20  
msgbox('B');  
    [y, Fs]=wavread('B.wav');  
sound(y, Fs);  
elseif ma > 20 && ma <=30  
msgbox('C');  
    [y, Fs]=wavread('C.wav');  
sound(y,Fs);  
elseif ma > 30 && ma <=40  
msgbox('D');  
    [y, Fs]=wavread('D.wav');  
sound(y,Fs);  
elseif ma > 41 && ma <=45  
msgbox('E');  
    [y, Fs]=wavread('E.wav');  
sound(y,Fs);  
elseif ma > 46 && ma <=50  
msgbox('F');  
    [y, Fs]=wavread('F.wav');  
sound(y,Fs);  
elseif ma > 51 && ma <=55  
msgbox('G');  
    [y, Fs]=wavread('G.wav');
```

```

sound(y,Fs);
elseif ma > 56 && ma <=60
msgbox('H');
    [y, Fs]=wavread('H.wav');
sound(y,Fs);
elseif ma > 61 && ma <=65
msgbox('MAY I KNOW YOUR NAME?');
    [y, Fs]=wavread('may.wav');
sound(y,Fs);
elseif ma >= 66 && ma <=70
msgbox('THANK YOU');
    [y, Fs]=wavread('thankyou.wav');
sound(y,Fs);
elseif ma > 71 && ma <=75
msgbox('WHERE IS THE NEAREST HOSPITAL?');
    [y, Fs]=wavread('where.wav');
sound(y,Fs);
elseif ma >= 76 && ma <=80
msgbox('PLEASE HELP ME');
    [y, Fs]=wavread('please.wav');
sound(y,Fs);
elseif ma >= 81 && ma <=85
msgbox('CAN YOU CONVEY THIS MESSAGE FOR ME?');
    [y, Fs]=wavread('can.wav');
sound(y,Fs);
elseif ma >= 86 && ma <=88
msgbox('I');
    [y, Fs]=wavread('I.wav');
sound(y,Fs);

```

```
elseif ma >= 89 && ma <=91
msgbox('J');
    [y, Fs]=wavread('J.wav');
sound(y,Fs);
elseif ma >= 92 && ma <=94
msgbox('K');
    [y, Fs]=wavread('K.wav');
sound(y,Fs);
elseif ma >= 95 && ma <=97
msgbox('L');
    [y, Fs]=wavread('L.wav');
sound(y,Fs);
elseif ma >= 98 && ma <=100
msgbox('M');
    [y, Fs]=wavread('M.wav');
sound(y,Fs);
elseif ma >= 101 && ma <=103
msgbox('N');
    [y, Fs]=wavread('N.wav');
sound(y,Fs);
elseif ma >= 104 && ma <=106
msgbox('O');
    [y, Fs]=wavread('O.wav');
sound(y,Fs);
elseif ma >= 107 && ma <=109
msgbox('P');
    [y, Fs]=wavread('P.wav');
sound(y,Fs);
elseif ma >= 110 && ma <=112
```

```
msgbox('Q');
    [y, Fs]=wavread('Q.wav');
sound(y,Fs);
elseif ma >= 113 && ma <=115
msgbox('R');
    [y, Fs]=wavread('R.wav');
sound(y,Fs);
elseif ma >= 116 && ma <=118
msgbox('S');
    [y, Fs]=wavread('S.wav');
sound(y,Fs);
elseif ma >= 119 && ma <=121
msgbox('T');
    [y, Fs]=wavread('T.wav');
sound(y,Fs);
elseif ma >= 122 && ma <=124
msgbox('U');
    [y, Fs]=wavread('U.wav');
sound(y,Fs);
elseif ma >= 125 && ma <=127
msgbox('V');
    [y, Fs]=wavread('V.wav');
sound(y,Fs);
elseif ma >= 128 && ma <=130
msgbox('W');
    [y, Fs]=wavread('W.wav');
sound(y,Fs);
elseif ma >= 131 && ma <=133
msgbox('X');
```

```
[y, Fs]=wavread('X.wav');  
sound(y,Fs);  
elseif ma >= 134 && ma <=136  
msgbox('Y');  
    [y, Fs]=wavread('Y.wav');  
    sound(y,Fs);  
else ma >= 137 && ma <=139  
msgbox('Z');  
    [y, Fs]=wavread('Z.wav');  
    sound(y,Fs);  
end
```

7. REFERENCES

- [1] V.I. Pavlovic, R. Sharma, T.S. Huang. Visual interpretation of hand gestures for human-computer interaction, A Review, IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7): 677-695, 1997.
- [2] J.Davis, M.Shah, Recognizing hand gestures, in Proceedings of European Conference on Computer Vision, ECCV: 331-340, 1994.
- [3] D.J.Turman, D. Zeltzer, Survey of glove-based input, IEEE Computer Graphics and Application 14:30-39, 1994
- [4] Starner, T. and Pentland, Real-Time American Sign Language Recognition from Video Using Hidden Markov Models, TR-375, MIT Media Lab, 1995.
- [5] R.Kjeldsen, J.Kender, Visual hand gesture recognition for window system control, in IWAFGR: 184-188, 1995.
- [6] M.Zhao, F.K.H. Quek, Xindong Wu, Recursive induction learning in-hand gesture recognition, IEEE Trans. Pattern Anal. Mach. Intel. 20(11): 1174.
- [7] Hyeon-Kyu Lee, Jin H. Kim, HMM based threshold model approach for gesture recognition, IEEE Trans. Pattern Anal. Mach.Intel.201(10):961-973,1999.
- [8] Ho-Sub Yoon, JungSoh, Younglae J. Bae, Hyun Seng Yang, Hand Gesture recognition using combined features of location, angle, velocity, Pattern Recognition 34:1491-1501,2001.
- [9] R.Locken, A.W. Fitzgibbon. Real gesture recognition using deterministic boosting , Proceeding of British Machine Vision Conference, 2002.
- [10] Hand Gesture Recognition Using Neural Networks by Klimis Symeonidis.
- [11] Duane Hanselman, Bruse Littlefield, Mastering MatLab, A comprehensive tutorial and reference, Prentice Hall.