**Faculty Name:**K.Chandra Sekhar

**Primary Research Area:** Image Processing

**Secondary Research Area:** Neural networks

**Project Title : Sign Language and Hand Gesture Recognition**

**What is Image Processing?**

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too.

Image processing basically includes the following three steps:

* Importing the image via image acquisition tools;
* Analysing and manipulating the image;
* Output in which result can be altered image or report that is based on image analysis.

There are two types of methods used for image processing namely, analogue and digital image processing. Analogue 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. Digital image processing techniques help in manipulation of the digital images by using computers. The three general phases that all types of data have to undergo while using digital technique are pre-processing, enhancement, and display, information extraction.

**WHAT IS SIGN LANGUAGE ?**

 A language that employs signs made with the hands and other movements, including facial expressions and postures of the body, used primarily by people who are deaf. There are many different sign languages as, for example, British and American sign languages. British sign language (BSL) is not easily intelligible to users of American sign Language (ASL). Unlike ASL, BSL uses a two-handed alphabet. In developing countries, deaf people may use the sign language of educators and missionaries from elsewhere in the world.

**WHAT IS SIGN LANGUAGE AND HAND GESTURE RECOGNITION?**

This project is aimed to convert the signs and gestures shown by the user into text. It bridges the communication gap between people who cannot speak and the general public. Image processing algorithms along with neural networks is used convert the raw images/videos into respective text that can be readable

**PROJECT DEVELOPMENT STAGES:**

## Skin Segmentation

The first step in implementing our particular gesture­recognition system is being able to effectively segment skin pixels from non­skin pixels. By using only a simple RGB­based webcam we are limited in methods for locating and distinguishing static hand gestures. For this reason, we have chosen to focus on segmentation using various color spaces. Skin segmentation methods are generally computationally inexpensive, and moreover, they can function robustly across many different models of simple webcams ­ an important feature given the notable difference in quality, color, etc. that can exist between webcams.

Specifically, we use a basic thresholding technique, choosing min and max threshold values which contain well­known and thoroughly tested skin pixel value regions. In total we use three different color spaces: 1) RGB (red­blue­green), 2) HSV (Hue­Saturation­Value), and 3) YCbCr (luminance, blue­difference and red­difference chroma). Using three color spaces is a computationally efficient procedure, especially given the added robustness it provides to distinguishing skin and non­skin values. HSV and YCbCr color spaces provide additional information about the separation between luminance and chrominance that RGB color spacedoesn’t provide. In all we use nine different pairings of min­ and max­ threshold values, each representing a given color space, then, applied certain boundaries rules introduced by Thakur et al. [2]. A particular pixel is part of skin region if and only if that pixel value passes boundaries rules.

## Hand Detection

The next step in implementing this static gesture recognition system is locating the hand in the video frame. At this stage we are provided a segmented skin mask of the original image. While the segmented skin mask provides us the regions in which the hand could be, the hand is only part of all that is segmented from the mask. Since the mask captures skin regions ­ most notably the arm and face skin ­ we must do more processing to find the hand. Specifically, we want to find an indistinguishable characteristic to the hand, namely a consistent feature we can always find. With that mindset, we felt the palm of the hand provided that consistency. Much of the difficulty in finding the location of the palm is the variability in size. We didn’t have any information regarding the position of the user’s hand as well as size (i.e. the closer the user’s hand to the screen the larger hand appears). In addition, we were more interested in fingers which played a crucial role in static hand gesture recognition algorithm (discussed in the next section).

To solve this problem, we took advantage of the hand geometry. Any human hand’s largest cross section area lies somewhere near the middle of the palm and independent of size. With this property, we use the centroid equation to find the arithmetic mean position of the hand which corresponds to the center of the palm. We can easily differentiate between the top and bottom of the hand after finding center of the hand. By using this method, we provided a robust algorithm to detect the center of any human hand.

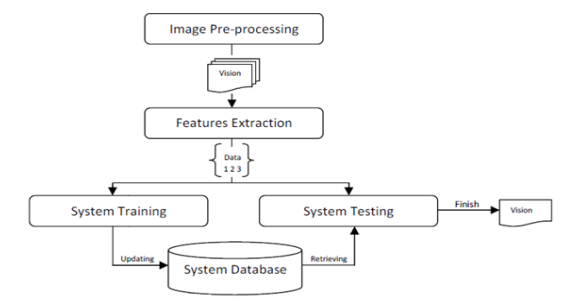
## Static Hand Gesture Recognition

After the location of the hand is found, the challenge becomes identifying different gestures of the given hand. The main difficulty in this is what type of features to use in distinguishing various static gestures. We concluded that most static gestures relied on the hand’s fingers. Though a simple conclusion, it is this very deduction that our algorithm was built around. While more complex algorithms for contour detection and gesture recognition exist, we chose a simpler algorithm based on and adapted from Malima [3].

In our adapted model, we draw a rectangle around the center of the palm found in the preceding procedure. This rectangle’s size is proportional to the furthest skin pixel value from the center of the palm. Ideally the rectangle drawn does not entirely encapsulate the hand.

Instead, the lines of the rectangle only contain the palm, cutting through the fingers and wrist. In this process we make a key assumption, in that the hand is upright (i.e. the line of the rectangle *that cuts through the fingers is known.) Then we simply calculate the number of fingers that cross the line of the rectangle (calculations explained in greater detail in the section below.) This algorithm provides us a very simple, computationally efficient method for counting the number of fingers any given hand has up.*

**Architecture**



**Prerequisites** to take up this project.

1. Image processing Tools i.e, openCV

2. Efficient programming language i.e. python or c++,

3. Algorithms.

4. Mathmatics behind the image processing.

***Referrence***

[**http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.454.3689&rep=rep1&type=pdf**](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.454.3689&rep=rep1&type=pdf)

[**https://pdfs.semanticscholar.org/7785/7eef494b30663e0c17270486f35851246e6d.pdf**](https://pdfs.semanticscholar.org/7785/7eef494b30663e0c17270486f35851246e6d.pdf)

<https://www.cse.unr.edu/~bebis/CS791E/Notes/EdgeDetection.pdf>