Sign language recognition using hand gestures

**INTRODUCTION**

With the increasing use of computing devices in day-to-day life, the need of user-friendly interfaces has led towards the evolution of different types of interfaces for human computer interaction.

Real time vision-based hand gesture recognitionaffords users the ability to interact with computers in more natural and intuitive ways.

Direct use of hands as an inputis an attractive method which can communicate much more information by itself in comparison to mic, joysticks etc allowing a greater number of recognition system that can be used in a variety of human computer interaction applications.

This can be very helpful for the deaf and dumb peoplein communicating with others as knowing sign language is not something that is common to all, moreover, this can be extended to creating automatic editors, where the person can easily write by just their hand gestures.

**GESTURE RECOGNITION**

The word gesture can refer to any non-verbal

communication that is intended to communicate a specific message.

Gesture recognition is a type of perceptual computing

user interface that allows computers to capture and interpret human gestures as commands. The general definition of gesture recognition is the ability of a computer to understand gestures and execute commands based on those gestures.

HEAD AND FACE GESTURES: When people interact with one another, they use an assortment of cues from head and to convey information. Some examples are knodding head, raising eyebrows, etc.

HAND AND ARM GESTURES: Majority of automatic recognition systems are for deictic gestures(pointing), emblematic gestures (isolated signs) and sign languages.

DATA GLOVES: Data gloves relay a user’s hand and fingers movements to a VR system, which then translates the wearers gestures into manipulations of virtual objects.

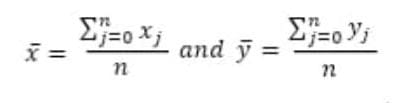
BODY GESTURES: It is a tracking of full body motion, recognizing body gestures, & recognizing human activity.

**HAND SEGMENTATION**

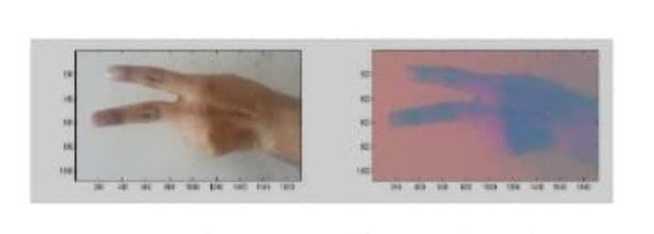
LOCALISING HAND REGION: Our 1st task is to segment out the hand in the video from the background. We achieve this result in two steps: We find the pixels that belong to the hand region. It is been observed that YCbCr color space gives better

clustering result & computational efficiency. So, we convert our image YCbCr color space.

FINDING CENTROID: we calculate centroid by using following formula,



After we obtain centroid, we find distance from most extreme point in the hand to center. We then draw a circle whose radius is farthest distance from centroid. This circle contains the whole gestures.



**FEATURE EXTRACTION**

For feature extraction, we need to understand neural network.

NEURAL NETWORKS: Neural Networks are composed of simple elements operating in parallel which are inspired by biological nervous systems. Commonly, neural networks are adjusted, or trained so that a particular input leads to specific target output.

Neural Networks have been trained to perform complex function in various fields of applications including pattern recognition, identification, classification, speech and vision and control systems.

**ROW VECTOR ALGORITHM**

For each type of hand gesture, we took several hand images, do skin modeling, labeling, removed their background and RGB to binary conversion in the preprocessing phase, calculated their row vectors and then trained the neural network with these row vectors. Ultimately, the neural network was able to recognize the row vectors that each gesture count can possibly have. Hence, after training, the system is tested to see the

recognition power it had achieved.

**EDGING AND ROW VECTOR PASSING ALGORITHM**

For representing the brightness of pixels there are two ways for represent numbers, first class called Double class that assign floating numbers (“decimals”) between 0 and 1 for each pixel.

After the conversion of the image into grayscale, we took the edge of the image with a fixed. Threshold which helped us in removing the noise in the image. In the next step, a row vector of the edged image was calculated. This row vector is then passed on to the neural network for training. The neural network (NN) is later on tested for the classification of the gestures.

**MEAN AND STANDARD DEVIATION OF EDGED IMAGE**

In the pre-processing phase, doing several steps like removing the background and RGB image is converted into grayscale type as done in the previous algorithm. The edge of the grayscale image is taken with a fixed threshold then calculate the mean and standard deviation the processed image. Mean is calculated by taking a sum of all the pixel values and dividing it by the total no of values in the matrix. Standard Deviation can calculate from mean which is mathematically defined as: The mean and standard deviation of each type of count gesture are given to the neural network for training.

**DIAGONAL SUM ALGORITHM**

The binary image format also stores an image as a matrix but can only color a pixel black or white (and nothing in between). It assigns a 0 for black and a 1 for white. In the next step, the sum of all the elements in every diagonal is calculated. The main diagonal is represented as k=0 the diagonals below the main diagonal are represented by k<0 and those above its Diagonal Sum are represented as k>0. The gesture recognition system developed through this algorithm first train itself with the diagonals sum of each type of gesture count at least once, and then its power could be tested by providing it with a test gesture at real time.

**TRAINING THE MODEL**

We will use Keras library for the training process. Keras is a simple and powerful python library that is widely used in deep learning. It has made it really easy to train neural network models.

Let us look at some of the parameters used for our training.

Sequential: We will use a sequential model, meaning that the layers are arranged in a linear stack (sequence). The layers in a model are added as arguments to this constructor. Nodes: Number of neurons/nodes in our output layer is equal to the number of classes we are trying to predict.

Input Shape: The input shape is at least 225 X 225 (with 3 channels for RGB). Activation Function: We are using the ReLU activation function, which is the most common activation function that is used in neural networks due to its computational simplicity (among other advantages). This function returns a zero for negative inputs and the value itself for positive inputs. For most of the modern neural networks, ReLU is the default activation function. After training we will store the trained model parameters into a file which we will be using later during model testing.

**TEST THE MODEL**

To test the model, we are using some images of hand gestures captured again with webcam. The program loads the model file and it takes the input image as a parameter and predicts the class it belongs to.

Before passing the image, we need to ensure that we are using the same dimensions that we used during the training phase. This is how we test the model.

**FAILURE ANALYSIS**

There are few factors which could decrease efficiency of system specially when we talk about real time classification problem occurs during training when one gesture end other begins, when user try to recognize gesture with different degree the system fails to recognize the gesture.

In different hand images the hand could have different orientation; this may affect hand recognition accuracy because when we get images from camera to testing and training distance of camera from hand and rotation of wrist makes difference, so it is essential to work with many degrees of freedom as possible in order to make recognition process realistic.

If we talk about orientation, we must deal this thing in processing phase doing framing of every image by finding main axis of hand and calculate orientation and reorientation.

**OUTPUT**

