# Hand Gesture Recognition using SVM.

**Manan Mittal**
**Sophomore**
**Department of Mechanical Engineering**
**Indian Institute of Technology, Delhi**
**me2190901@iitd.ac.in**

## Abstract

For the communication between human beings and in sign language, hand gestures play a significant role. In-fact Deaf people survive their life by communicating with hand gestures. But recognizing these gestures is a difficult task for those who do not know sign languages. With the emerging field of machine learning, we can take the help of computers to perform this task. In this paper, we propose a hand gesture recognition method, which is based on a support vector machine (abbreviated as SVM) classifier. We will use Canny's edge detection for detecting edges of the hand. After detecting edges, we will extract features of an image by Histogram of Gradients(HoG). And then finally classify these images based on features using Support Vector Machine. A model trained with an SVM classifier predicts whether a new sample falls in one category or the other.

## 1. Introduction

The continual improvement in multimedia technologies and the increased focus on image data, has paved way for impossible applications just a few years ago. Earlier, the interaction with computers was majorly limited to keyboard and mouse. But these days different methods of interaction with devices are also becoming possible. One such method is gesture recognition that has become possible because of ML (Machine Learning).

A gesture is a form of communication, non-verbal in nature, in which bodily actions are used instead of speech to communicate. Gestures can be static or real-time (sequence of gestures). Gesture recognition is way computers begin to understand human body language and it acts as a bridge between humans and computers. It is the most natural way of interacting with machines without any mechanical devices. Multiple ways have been explored for acquiring necessary information for gesture recognition system. Some methods use hardware devices like data glove devices and colour markers to extract the features of gesture easily. Other methods use skin color to segment hand and extract necessary features.

There are two types of hand gesture recognition popular these days: - Glove based hand recognition which requires devices to be worn and vision-based hand recognition in which features are extracted of visual appearance of input. In this paper we are implementing vision based hand gesture recognition which has benefit of being probe less and can work on almost any computer with decent web-cam.

## 2. Objective

We are using Support Vector Machine (SVM) classifier that recognises hand gestures and then classify them. The images are first pre-processed using Canny's edge detection and Contour maps. Using these methods, we remove noise by detecting edges, removing unwanted pixels by checking if the pixel is a local maximum in its neighbourhood in the direction of gradients and then finally use hysteresis threshold. In the next step we use HOG to extract features of the image which includes computing horizontal and vertical gradients with the help of which we can compute gradient direction and magnitude matrices. Using these two matrices we compute histogram of oriented gradients. The extracted features are then trained using SVM which classifies the gesture and testing is performed. SVM is a popular supervised learning pattern recognition technique. It constructs a hyper plane, which is constructed as a decision surface, in such a way that the margin of separation between different classes is maximized.

## 3 Approach

We will take input images from the webcam and then convert RGB to Greyscale colour space and some resizing can be also done for faster computation**.** Then we will detect the edges of the image with the help of Canny's edge detection method. After detecting edges of the image, we can use Histogram of Gradients(HoG) to extract features from the image. These features can be stored in the database. Then we will use SVM to classify the feature by matching it with the most similar feature in the trained database.

For training of the model, we will produce our own dataset which will contain labelled images for each class (0,1,2,3,4,5) and will test and compare on trained model.

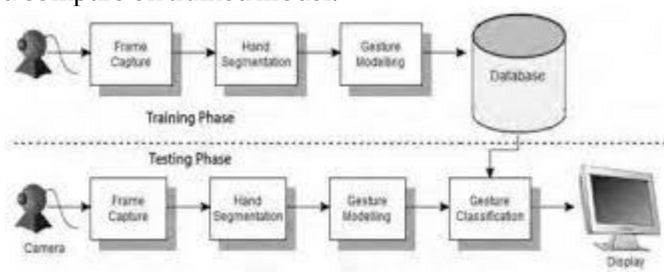



Fig.1 Project Model

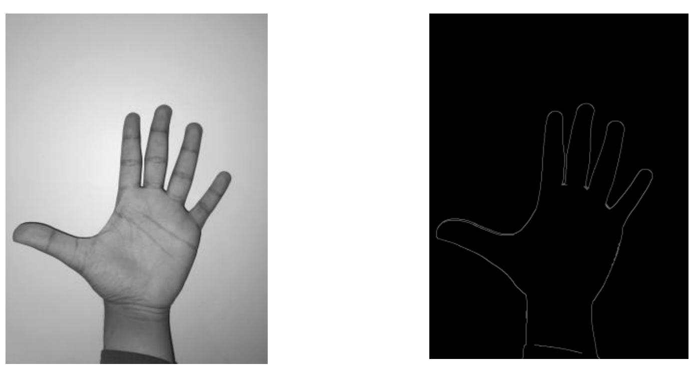
### 3.1 Canny's Edge Detection

Canny's Edge Detection technique is one of the most used and efficient edge detection methods used in image processing. The point at which image brightness changes sharply are typically organized into a set of curved line segments termed edges. The purpose of Edge detection in general is to significantly reduce the amount of data in an image while preserving the structural properties to be used for further image processing.

Steps for edge detection are: -

1. Smoothing.
2. Finding Gradients.
3. Non-Maximum Suppression.
4. Edge Tracking by Hysteresis.

***(Note: - More details on these steps are included in supplementary material file)***



**Original Image** **Edge Detected Image**

Figure 2: Example of edge detection using Canny's Edge Detection.

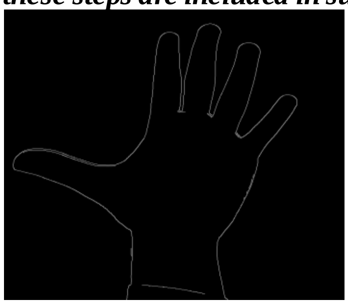
### 3.2 Histogram of Oriented Gradients

HOG, or Histogram of Oriented Gradients, is a feature descriptor that is often used to extract features from image data. HOG features were introduced by Navneed Dalal and Bill Triggs. It is a type of dimensionality reduction that captures part of images as compact feature vector. When it comes to detection to human features even in cluttered backgrounds and under difficult illumination, locally normalized Histogram of Oriented Gradient (HOG) descriptors provide excellent performance relative to other existing feature sets including wavelets. In the HOG feature descriptor, the distribution (histograms) of directions of gradients (oriented gradients) are used as features. Gradients (x and y derivatives) of an image are useful because the magnitude of gradients is large around edges and corners (regions of abrupt intensity changes) and we know that edges and corners pack in a lot more information about object shape than flat regions.

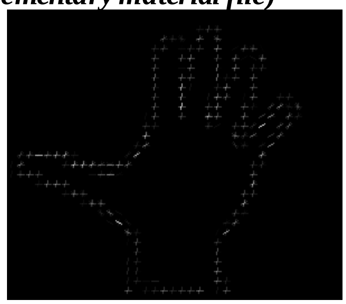
HOG algorithm includes three steps:-

1. Gradient Computation.
2. Orientation binning.
3. Descriptor Blocks.

***(Note: - More details on these steps are included in supplementary material file)***



**1.Canny Edge Detected image**



**2.Visualization of HOG**

### 3.3 Classification using Support Vector Machine

Support Vector Machine (SVM) is one of the best machine learning algorithms, which was proposed in 1990‟s by Vapnik. SVM‟s are a set of related supervised learning methods used for classification and regression. A supervised learning algorithm analyses the training data and produces an inferred function which is called a classifier. SVM has many advantages over other classifiers as it provides higher accuracy, guarantees regarding overfitting, and provides an appropriate kernel that can work well even for which is non linearly separable in the base feature space.

SVM constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. A hyperplane is a subspace of one dimension less than its ambient space. A good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class. It is explicitly told to find the best separating hyperplane. It searches for the closest points which are called the "support vectors". Once it has found the closest points, the SVM draws a line connecting them. It draws this connecting line by doing vector subtraction (point A - point B). The support vector machine then declares the best separating line to be the line that bisects and is perpendicular to the connecting line.

In this project we will use SVM to do multiclass classification which will use Linear kernel function to avoid overfitting.

**Example explaining SVM:**

Let us imagine we have two tags: green and blue, and our data has two features: x and y. We want a classifier that, given a pair of (x, y) coordinates, outputs if it is either green or blue. We plot our already labelled training data on a plane:

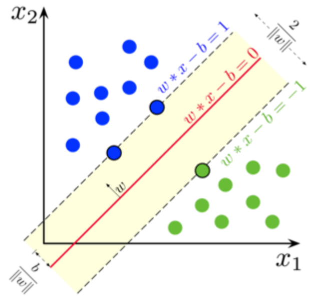



Figure 4: Showing Decision Boundary created by SVM between different datasets.

A support vector machine takes these data points and outputs the hyperplane (which in one dimension less than its ambient space is simply a line) that best separates the tags. This line is the decision boundary: anything that falls to one side of it we will classify as blue, and anything that falls to the other as green.

The specialty of SVM is when you get a new sample or new data point you have already made a line that keeps B and A as far away from each other as possible, and so it is less likely that one will spill over across the line into the other's region.

### 3.4 Forming Training and Testing Dataset.

We will make our own dataset by capturing images from our webcam and labelling them as 0,1,2,3,4,5 specific to their class. Then we will divide our dataset into training and testing dataset. For making our model more accurate we ensure that our dataset will contain maximum cases of occurrence for each class. So, we will capture images from a low-medium quality camera in a varied light conditions, different angles, and spaces between fingers. As SVM is based on deterministic algorithm we will take large dataset for training and testing, our training dataset will contain about 350 images for each label and testing dataset will contain over 100 images for each label.

## 4 Results

### 4.1 Comparing Image Pre-Processing Techniques.

Firstly, we reduce our area of coverage as compared to original frame to capture only the part containing hand and no other objects and draw a rectangle on original frame around that area for convenience. Then we converted each image to grayscale. After that, we use various noise reducing techniques like contour retrieval, gaussian-blur, masking with skin colours as threshold and much more but the best results are obtained when we apply Canny edge detection on converting original images to grayscale images. Using canny edge detection, we obtained an image with least noise, it automatically detects edges of hand and differentiate between hand and its shadow in a single frame it is also insensitive to distinct colours of hand and background. But for best results you must take plane background otherwise there will be some disturbance in detected edges.
You can compare various noise reduction techniques by samples given below.

**1.Original Image**

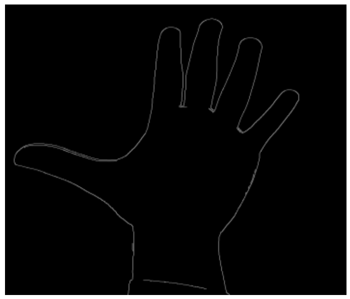
**2.Skin Colour Threshold Image**

**3.Skin Colour Masked Image**

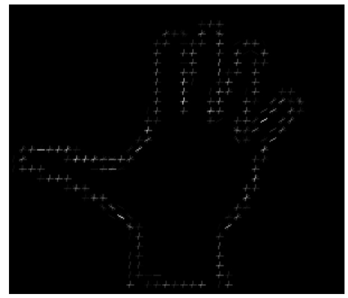
**4.Canny Edge Detected Image.**

### 4.2 Extracting Feature from Edge detected images.

We use Histogram of Oriented Gradients (HOG) to extract features from Edge detected image. HOG and canny edge detection have common initial steps such as filtering and calculating image gradients, so by combining them some computation time is reduced. HOG can extract region of interests in an image by calculating gradients. We applied HOG on each of the edge detected image and stored all of them in a 2-D vector containing extracted features from image.



**1.Canny Edge Detected image**



**2.Visualization of HOG**

### 4.3 Implementing SVM classifier on extracted features.

Our dataset contains labelled images for different Hand gestures. For example: - For image showing gesture 1: image name is saved in folder named 1. Like that we have six folders corresponding to all six classes i.e., 0,1,2,3,4,5. We trained SVM model with 'Linear Kernel' to classify these images features (extracted with HOG) between different classes. Using linear kernel in multiclass classification has benefits of reducing overfitting. We will make use of OneVsRest type SVM classifier to facilitate multiclass classifications. Then we test our model with testing dataset.

Results corresponding to each class is shown in table below.

| Gesture Label | Accuracy of predicting correctly in 100 test samples. |
|---|---|
| 0 | 76% |
| 1 | 93% |
| 2 | 97% |
| 3 | 79% |
| 4 | 100% |
| 5 | 81% |
| **Overall** | **87.66%** |

By above results we can say that overall, our model is giving satisfactory results. For our testing dataset, accuracy for predicting 4 is quiet high which may happen when all the samples in testing dataset lies within the region for that class or we can say that the decision boundaries are made such that they classify gesture 4 accurately but results can be different for other datasets. For gesture 0 and 3 accuracy is low with which we can say that decision boundary is not able to classify these gestures perfectly. Although these results can be different for different datasets and kernel functions but as we have taken large amount of data for training and testing with different angles of gestures and lighting condition, we can say that our model have less chances of facing sparse data problem.

### 4.4 Implementing our model for classifying live inputs from webcam.

We applied a method based on contour area to detect whether an object is in frame or not. If there is no object in the frame, the method will not permit input to go for prediction, this will make our model work faster by avoiding model to classify unnecessary inputs. Then, we tested our model to classify inputs of different gestures from webcam and observed that model is able to correctly recognize 1,2,4 and 5 in a greater frequency than 0 and 3. This observation is consistent with the results we obtained when we run test on static images during testing phase.

**(Note: - Demonstration for this part is included in presentation video.)**

# 5 Conclusion

Area of Hand gesture-based computer human interaction is very vast. This project recognizes hand gesture offline. So, work can be done for real time purpose. Best part of our model is: - we can deploy this on any computer, even with low to medium quality webcams it works well. Hand recognition system can be useful in many fields like robotics, computer human interaction and so make this offline system for real time will be future work to do. In this project we used SVM for multiclass classifications but as we know SVM is deterministic algorithm and is not much accurate for multiclass classifications we can use other classifiers such as RVM, which is more powerful than SVM as it is based on probabilistic algorithm. We can also use neural networks to obtain better accuracy but implementation of both RVM and neural networks require knowledge of higher-level mathematics and coding.

Accuracy and functionality of Gesture recognition models also depends on external factors like input equipment, background, lighting conditions and many more. Rigorous research has been carried out in the field of image processing to lower these effects and enhance Human-computer interactions in last 20 years. Special cameras like depth sensor camera and IR sensing camera have been developed, that will provide much more information from an image than an ordinary camera. Although none of the model can provide us 100% accurate results lots of research has been done to increase accuracy of computer vision-based methods. To continue this momentum, it is clear that further research in the areas of feature extraction, classification methods and gesture representation are required to realize the ultimate goal of humans interfacing with machines on their own natural terms.

**References**


[1] https://learnopencv.com/histogram-of-oriented-gradients/

[2] https://medium.com/analytics-vidhya/a-gentle-introduction-into-the-histogram-of-oriented-gradients-fdee9ed8f2aa

[3]https://medium.com/@nikatsanka/comparing-edge-detection-methods-638a2919476e#:~:text=Canny%20Operator,then%20the%20ones%20described%20above.

[4] Michahial, Stafford. (2015). Hand gesture recognition using support vector machine. The International Journal Of Engineering And Science. 4. 42.

[5]https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/#:~:text=A%20support%20vector%20machine%20(SVM,able%20to%20categorize%20new%20text.

[6] S Nagarajan, T S Subashini, "Static Hand Gesture Recognition for Sign Language Alphabets using Edge Oriented Histogram" in the proceeding of IJCA(2013), Vol 82, No 4.

[7] S. Naidoo, C.W. Omlin M. Glaser "Vision based Static Hand gesture recognition", South Africa.