INDIAN INSTITUTE OF INFORMATION TECHNOLOGY ,GUWAHATI

PROJECT TITLE

Digit Detection via Motion of Head using SVM ,sklearn and openCV

SUBMITTED BY:

PRAGYANSHU VERMA , 1601039
DEPARTMENT OF ELECTRONICS AND COMMUNICATION

ABSTRACT :

This project is all about how to detect any digit via motion of the head according to shape of that digit. For this purpose, I have used simple camera of laptop to took the input. After running the program you need to move your head towards the shape of digit in the front of the camera. Here I used Dlib library, this library pretrained on human face detection(facial landmarks), after using this library we got centeral point of head (top point of nose). We track the path of that centeral point and store all coordinates, by using these coordinates we made a image and do some morphological operations, for better and clear image, like opening, closing, dilation, erosion etc. In the final step we send this image as a input to our digit predicting program that detect the digit (using linear svm method) and print output on terminal.

INTRODUCTION:

In this part we are going to intoduce description of main steps that had been taken in this project. First step is how to detect point on face, for this purpose, I used Dlib library (this library detects the facial landmarks). Detecting facial landmarks is a subset of face detection problem. Given an input image (and normally an ROI that specifies the object of interest), a shape predictor attempts to localize key points of interest along the shape.

In the context of facial landmarks, our goal is detect important facial structures on the face using shape prediction methods.Detecting facial landmarks is therefore a two step process:

Step #1: Localize the face in the image.

Step #2: Detect the key facial structures on the face ROI.

Face detection (Step #1) can be achieved in a number of ways.

We could use OpenCV's built-in Haar cascades.

We might apply a pre-trained HOG+ linear svm object detector specifically for the task of face detection.Or we might even use deep learning-based algorithms for face localization. In either case, the actual algorithm used to detect the face in the image doesn't matter. Instead, what's important is that through some method we obtain the face bounding box (i.e., the (x,y)-coordinates of the face in the image).

Given the face region we can then apply **Step #2: detecting key facial structures in the face region.This method starts by using:**

1. A training set of labeled facial landmarks on an image. These images are manualy labeled, specifying specific **(x,y)**-coordinates of regions surrounding each facial structure.
2. Priors, of more specifically, the probability on distance between pairs of input pixels.

Given this training data, an ensemble of regression trees are trained to estimate the facial landmark positions directly from the pixel intensities themselevs (i.e., no “feature extraction” is taking place). In this facial detection we are focusing on point number 34.

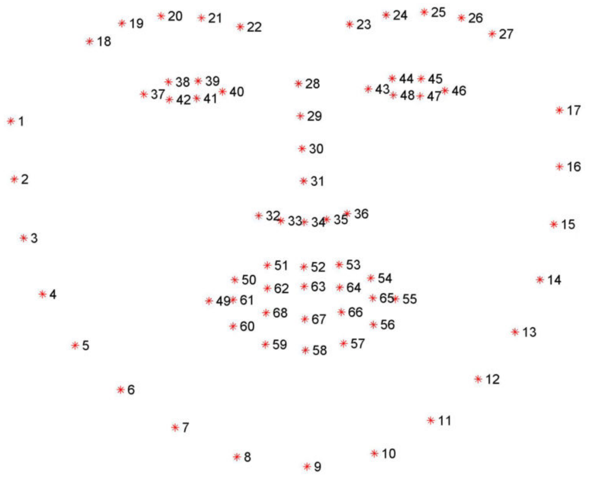


Fig.(i): facial landmarks

Fig.(ii):original output image of 1(one)

We tracked the path of the point 34 [fig(i)] and stored all coordinates of it and made a image [fig (ii)] by using these points. After looking at the final (output) image [fig(ii)], output image was not so clear that is why we did some morphological operations.

fig(iii):

fig(iv):

Image of the fig(iii) and fig(iv) are the image of after erosion and opening operation of image of fig(ii) respectively.It is clear by above figure that the image after erosion can be a good image for predicting. Now we moving the next step that is how to predict this digit. I did digit prediction using HOG feature. The **histogram of oriented gradients (HOG)** is a feature descripter 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 invarient 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.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 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 HOG descriptor is thus particularly suited for human detection in images.This image represents rough idea of HOG feature. See image on the side.

You will notice that dominant direction of the histogram captures the shape of the person, especially around the torso and legs.

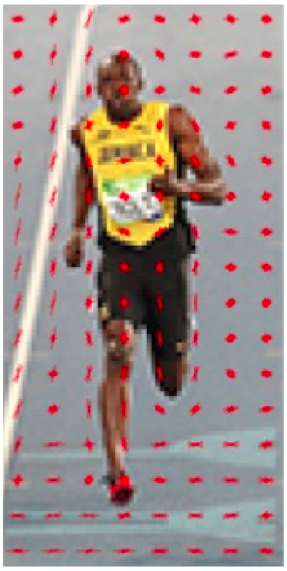
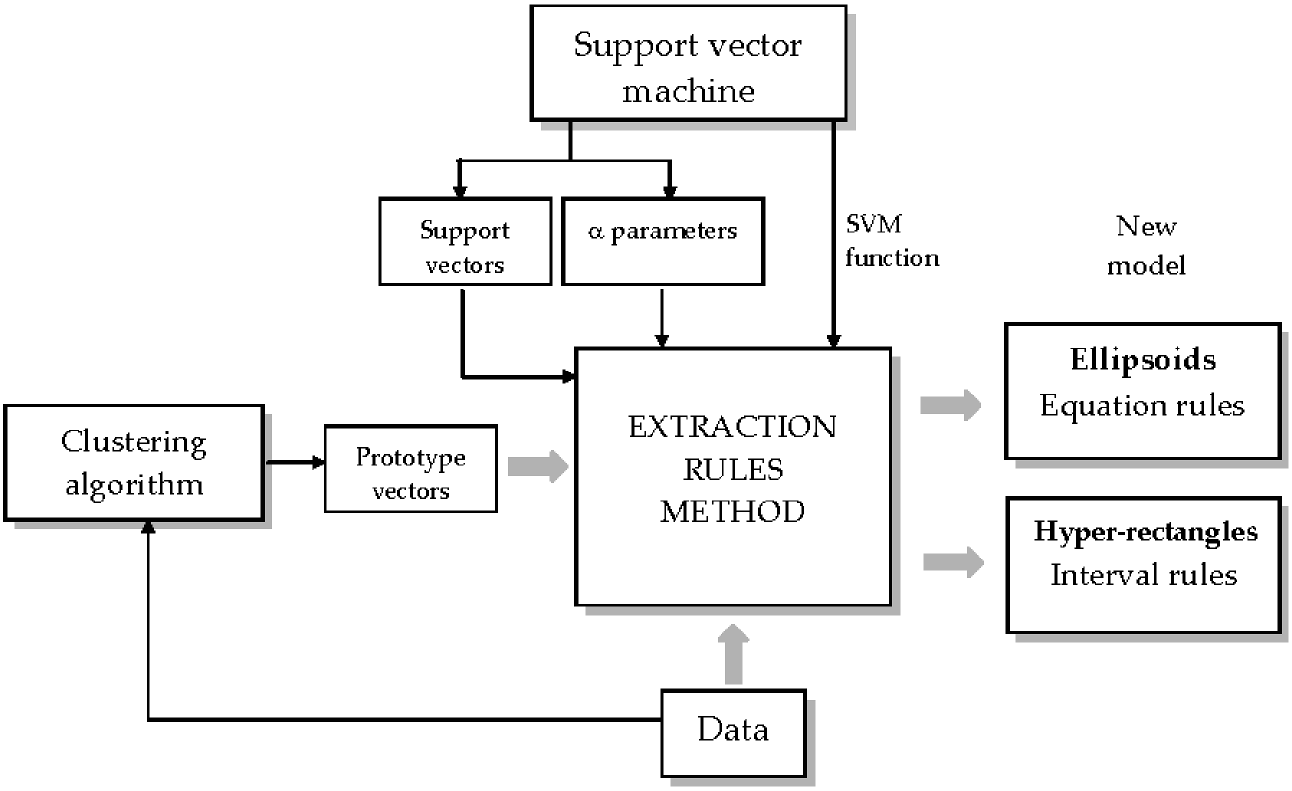


Fig.(v): visualizing hog

Now the next step is to detect the digit of image (that looks like handwritten digit), for this purpose I used linear svm method and predict the digit. To train the classifier I used the MNIST data base of handwritten digits. The basic steps are below-first is Create a database of handwritten digits. Second for each handwritten digit in the database, extract HOG feature and train a Linear SVM and in last use the classifier trained for digit prediction.A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimentional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithm that analyze data used for classification and regression. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.Kernel methods owe their name to the use of kernel functions, which enable them to operate in a high-dimensional, implicit feature space without ever computing the coordinates of the data in that space, but rather by simply computing the inner products between the images of all pairs of data in the feature space. This operation is often computationally cheaper than the explicit computation of the coordinates. This approach is called the “kernel trick”. Kernel functions have been introduced for sequence data, graphs, text, images, as well as vectors.



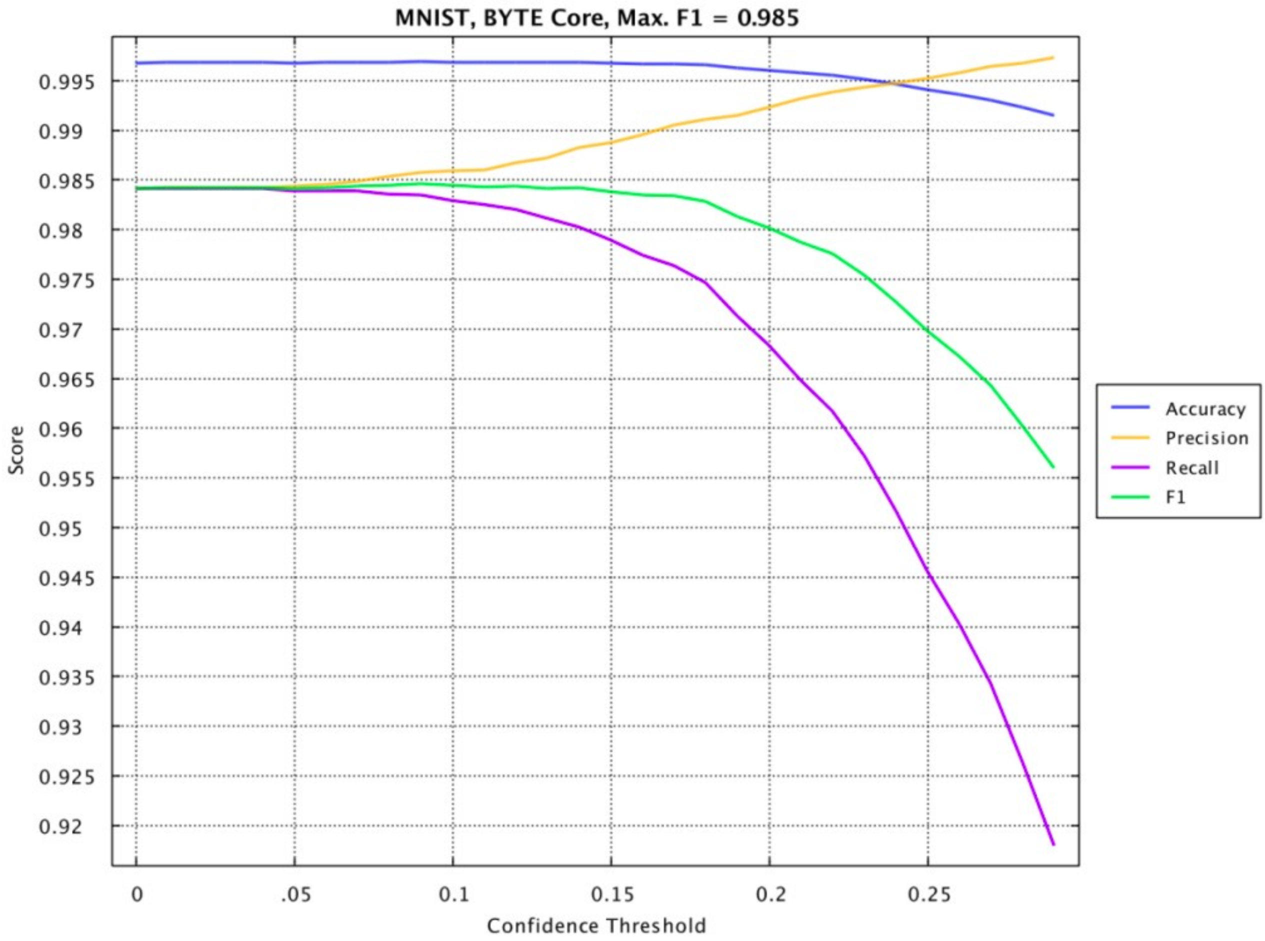
EQUATIONS:

we can use svms with Gaussian kernel on datasets that are not linearly separable. To find non-linear decision boundaries with the SVM, we need to first implement a Gaussian Kernel. The Gassian kernel is a similarity function that measures the "distance" between a pair of examples (x,y). The Gaussian Kernel is also parameterized by a bandwidth parameter, (sigma), which determines how fast the similarity matric decreases.

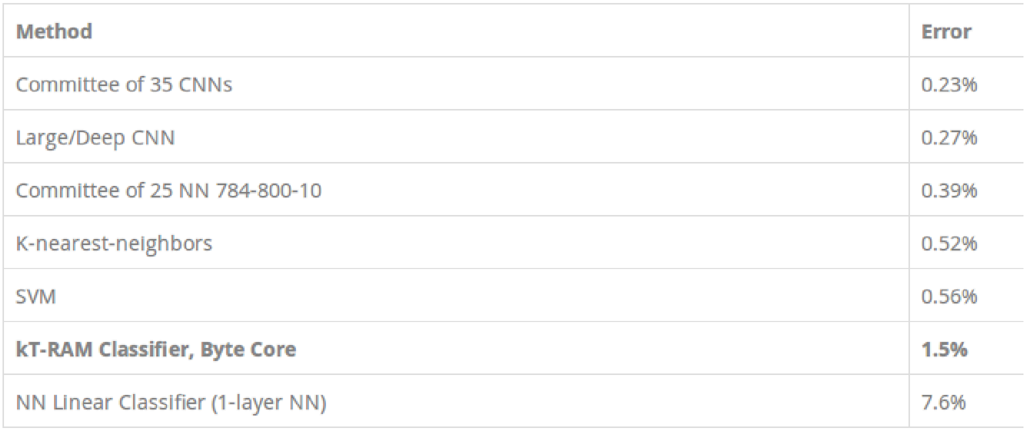
$$K_{gaussian}(x^{(i)}, x^{(j)}) = \exp\left(-\frac{\|x^{(i)} - x^{(j)}\|^2}{2\sigma^2}\right) = \exp\left(-\frac{\sum_{k=1}^{n}(x_k^{(i)} - x_k^{(j)})^2}{2\sigma^2}\right).$$

GRAPHS: visualization of MNIST database

The chart below displays the performance results as we vary our evaluation threshold. We manage to attain a peak classification error rate of 1.5% over the full dataset. As is commonly done, we use the overall dataset precision to evaluate error. $Error = 1 - \frac{TP}{TP+FP}$, where TP: = True positives, and FN = False negatives. In other words, error equals the number of digits falsely classified over the size of the test set.

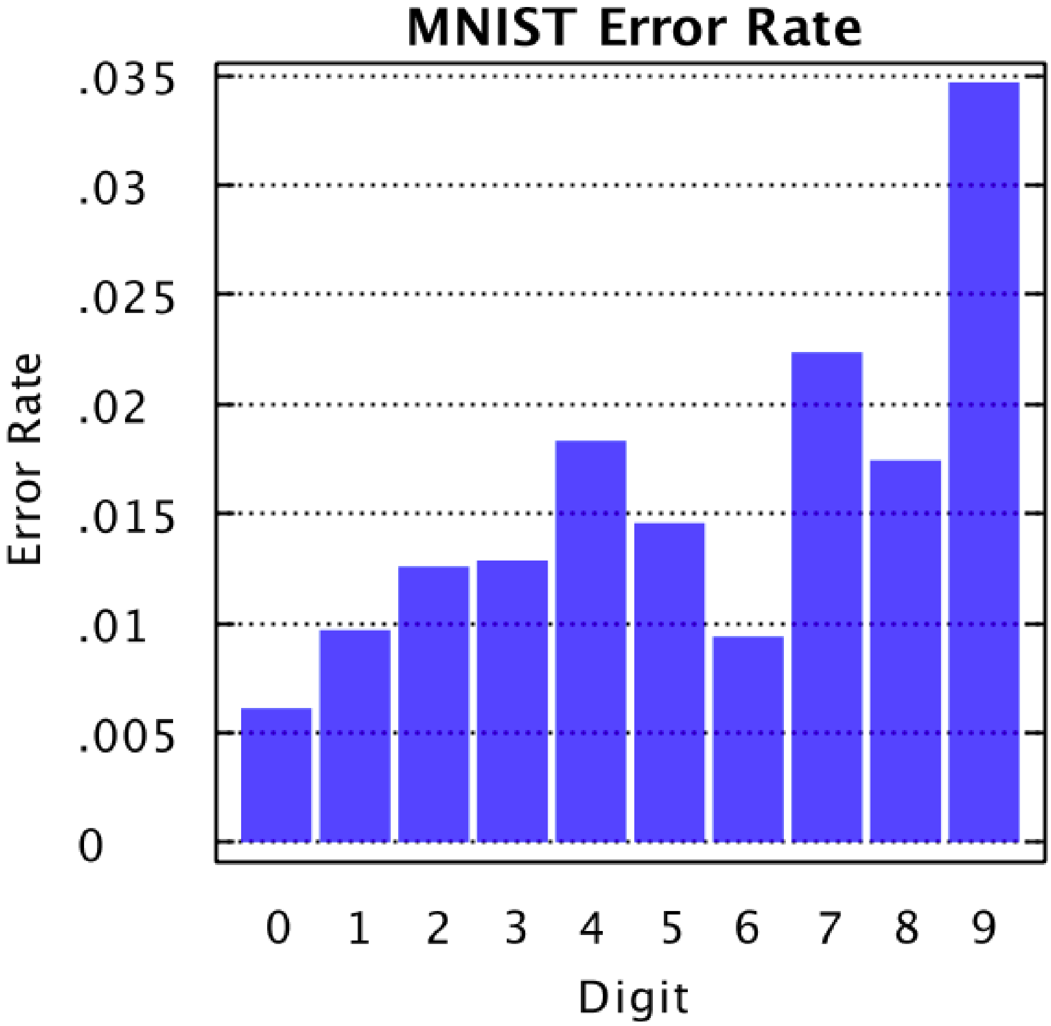


Given table shown error on MNIST data of different algorithm



Method	Error
Committee of 35 CNNs	0.23%
Large/Deep CNN	0.27%
Committee of 25 NN 784-800-10	0.39%
K-nearest-neighbors	0.52%
SVM	0.56%
kT-RAM Classifier, Byte Core	**1.5%**
NN Linear Classifier (1-layer NN)	7.6%

The following figure shows the error rate for the individual digits. We see that the toughest digit to recognize was "9".



CONCLUSION:

By using the concept of this project we can make virtual keyboard and can do other important stuff with the help of this. In the gaming it can be replace joystick and other high featured controller. Leap motion is an American company that manufactures and markets a computer hardware sensor device that supports hand and finger motion as input, analogous to a mouse, but requires no hand contact or touching. In 2016, the company released new software designed for hand tracking in virtual reality. The Leap Motion controller is a small USB peripheral device which is designed to be placed on a physical desktop, facing upward. It can also be mounted onto a virtual reality headset. Using two monochromatic IR cameras and three infrared LEDs, the device observes a roughly hemispherical area, to a distance of about 1 meter. The technology of leap motion is much advanced and also expensive, price of devices (leap motion devices) is very high. If we extend this project upto a level as leap motion then prices of devices and hardware part will get down. The table in the graph section represents error in different algorithm so there are many different algorithm by which accuracy of this project (or program) can be increases.

REFERENCES:

1. Joseph Howse-OpenCV Computer Vision withPython.
2. An Introduction to support vector machine and other kernel-based learning Methods-By John Shawe-Taylor and Nello cristianini
3. Morphological Image analysis by Soille,pierre
4. Digital Image processing by Rafael C. Gonzalez, Richard E. Woods