

Progress Report

On

Interacting with software using gesture recognition

Team Members

153050002

153050023

153050045

153050051

March 21, 2016

Contents

1	Method for recognizing gestures	3
1.1	Haar Classifier	3
1.2	Cascade Haar classifier	3
2	Using opencv to train and test Haar classifier	4
3	future Work	5
4	Pointers to literature	5

1 Method for recognizing gestures

We are using **Haar** classifier for Hand Recognition, i.e., when the user comes in front of the camera we use the classifier to recognize the hand in the image captured using the Web camera.

After successful identification of the area containing the hand we use that subimage for further classification, i.e., we have several fixed gestures mapping to several fixed software actions and we classify the coming input image as one this fixed classes of gestures and take the action corresponding to the gesture, e.g., scrolling the pdf file.

1.1 Haar Classifier

It is a famous classification algorithm for identifying particular objects in images in an efficient manner.

Haar Classifier considers adjacent rectangular regions at specific locations present in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums. The detection window can be moved throughout the image as the object can be located anywhere. This difference is then used to categorize subsections of an image. In our case the Haar features try to identify the intensity difference between the background and the region of our hand. These differences follow certain pattern because eventhough the possibilities are large but our hand still has predictable shape and size.

1.2 Cascade Haar classifier

The number of Haar classifier turn out to be very large even for small images and hence if detection is required in real time this results in efficiency barrier. So instead of applying the whole classifier at once, it is broken down into a series of weak classifiers and they are tested sequentially. Earlier stages contain less number of features and hence can be used to prune the area that does not have the object(hand). So further classifier need not be run on the windows failing the previous classifier.

2 Using opencv to train and test Haar classifier

We have used the opencv library to train the Haar cascade classifier for detecting hands. It is explained in the below stages:-

- **Collecting negative and positive images:**

We have used the `opencv_createsamples` command to build a set of positive examples by taking a single image of a hand and several background images and inserting the hand image into the background image at random locations angles and scales. And we kept the background images as the negative images.

Command:- `opencv_createsamples -img positive.jpg -bg bg.txt -info info/info.lst -pngoutput info -maxxangle 0.5 -maxyangle 0.5 maxxangle 0.5 -num 180` . This command outputs an `info.lst` file that contains the bound of the rectangle containing the object of interest.

- **Vectorizing the positive images:**

We have used below command to vectorize the image, i.e, converting them into a `.vec` file, which is a binary feature file for the images.

Command: `opencv_createsamples -info info/info.lst -num 180 -w 50 -h 50 -vec positive.vec`

- **Training the Haar classifier:** In this step we have used `opencv_traincascade` command to train the Haar classifier mentioning number of stages in the process and number of negative and positive images used in the training.

Command: `opencv_traincascade -data data -vec positive.vec -bg bg.txt -numPos 180 -numNeg 180 -numStages 5 -w 50 -h 50`

The Last step is combining the `.XML` files generated as a result of training the Haar classifier and combining them into a single `.XML` file representing the whole decision tree.

We have used the final `.XML` file in a sample python program provided by them and tested the classification and tested the classification on live video and the results are satisfactory, as the number of training examples were not enough.

3 future Work

We are planning to train the classifier on a large dataset say 2000 images and for various types of static gestures. We will then once again use the Haar classifier to further classify the detected hand into one of these static gestures and then build a small software showcasing the mapping from gestures to actions in the software.

4 Pointers to literature

http://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html

https://en.wikipedia.org/wiki/Haar-like_features

http://www.ermt.net/docs/papers/Volume_2/issue_3_March2013/V2N3-123.pdf

http://eeeweba.ntu.edu.sg/computervision/people/home/renzhou/Ren_Yuan_Zhang_MM11short.pdf