Visual Assistant using OpenCV

Project Report

Submitted by

Raghunath Reddy Jangam
&
Baozhi Yu

In partial fulfillment of the requirements for the course

Computer-vision -5722

Contents

CHAPTER 1
INTRODUCTION

1.1 Project Inspiration

How often do you swipe left on the phone to revisit the memories? Most of captured images are when we are happy. We both are definitely in the group who does that. I believe most of the funny/happy moments remain uncaptured because no one is aware that it is going happen. We want to do our part in helping people capture these moments and cherish them later on. Initially we started out with a plan to use an embedded development board to do this but the most of the time was allocated to make the board work which was not main concentration of the course. So we tweaked our idea and introduced other functionalities to make the system smart and act has a visual assistant.

We divided the visual assistant work into three modules

1.Color recognition and tracking

2.Gesture recognition

3. Face, eyes, smile detection

1.2 Color recognition & Tracking

In real world, computer can also interact with you by recognizing color. In this paper, we made two applications for a user to interact with computer via camera. First, a recognition of your shirt color is implemented. In other words, computer can identify which color you are wearing within the preset color. In this particular implementation, we assume the background is white and the preset colors are : red , orange, yellow, green, cyan, blue, purple and black. Second, a interaction can be implemented between user and computer by tracking a target color. We assume the background is white and the target color is red in this implementation. Four operations are defined by tracking the movement the target color object(maybe a red glove).

1.3 Hand & Gesture recognition

Based on the previous work on 1.1, we interact with computer by tracking target color. However, this approach has a limitation that we need to bring a object with target color all the time. To avoid this limitation, we explored and improved color tracking algorithm by recognizing human skin and draw a contour of human hand. Then, the center of maximum connected domain will be calculated and reported in the screen. Four operation are defined by tracking the difference of largest connected domain center frame by frame.

1.4 Smile detection

The main focus of this module will be on smile detection though we are going to go through face, eyes detection. The overall purpose of this module to guess mood of the user and make suggestions. In this module we consider two datasets one from Yale university and other from the real time camera.

1.4.1 Main goals

1.Segregate dataset of random images into Images with faces and without faces.
2.Using the dataset of images with the head shots and segregate the images into images with eyes.
3.Using the dataset of images with the head shots and segregate the images into images with smile.
4.In real time smile tracking, able to detect the smile from the frame grabbed from the camera.

CHAPTER 2
BACKGROUND THEORY

2.1 Module 1-Color recognition & Tracking

HSV color model:

HSV (Hue, Saturation, Value) is a color model which defines a certain color by different values of hue, saturation and value. The model of HSV is like a upside-down cone.

Hue is a factor measuring the relative degree from center axis. For example, color "blue" is around 240 degree. And the hue value of blue is 100-124 (in 0-180 scale).

Saturation is a factor measuring how strong the color is. In the figure above, saturation stands for how far a point from center axis is. For example, if the value of saturation for a point is low and it is located in the degree of blue, it means that the color of this point is dark blue. If the saturation value is high, it will be a light blue.

Value is a factor measuring the brightness. In the figure below, value stands for the height of a point. The brighter the environment is, more colors would be distinguished.

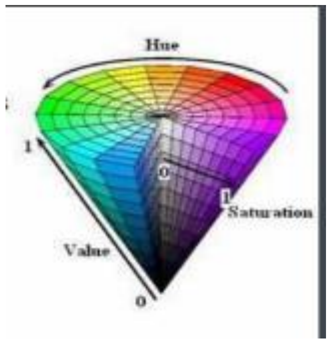


Figure 1 : HSV model

The setting of HSV for all preset colors are as follow:

Color	H	S	V
Red	156-180	43-255	46-255
Orange	11-25	43-255	46-255
Yellow	26-34	43-255	46-255
Green	35-77	43-255	46-255
Cyan	78-99	43-255	46-255
Blue	100-124	43-255	46-255
Purple	125-155	43-255	46-255
Black	0-180	0-255	0-46

Table 1 : Preset color threshold setting

Note that the boundary should be adjusted under different situations such as different brightness. More detailed color like violet can be identified with tighter bound. For example, the HSV value for violet is: 136,80,88. However, a detailed color would take more efforts in setting those bounds. So in this paper, we only work the 8 preset colors mentioned above.

Partial important functions:

1.cvtColor(colorimg,imgHSV,COLOR_BGR2HSV): Transform RGB input image colorimg to HSV format and return a output image imgHSV

2.Split(imgHSV,hsvsplit): Split HSV format image imgHSV to 3 channels. hsvsplit[0] is H channel. hsvsplit[1] is S channel. hsvsplit[0] is V channel.

3.inRange(imgHSV,scalar(iLowH,iLowS,iLowV),scalar(iHighH,iHighS,iHighV),imgthresholded):

Scan all the pixels in input image imgHSV and judge every pixel if it is has HSV value which is in the interval [iLowH,iHighH],[iLowS,iHighS] and [iLowV,iHighV]. If a pixel is inside those intervals, return a value of 255(white) in output image imgthresholded.

More morphology filter functions are also used in this part which will be discussed in detail

2.2 Module 2-Hand & Gesture recognition

There are two ways to reduce the noise as follow.

2.2.1 morphology filter

An important term in mathemetical morphology is structure element, the definition of which is a window doing mathematical morphology operations with pixels. There are 4 mathemetical morphology operations: erode, dilate, open and close.

Erode: Remove unrelated details in the image.

Dilate: recover the crack in the image.

Open: First erode, then dilate. Disconnect the narrow discontinuity and remove elongated protrusion.

Close: First dilate, then erode. Remove narrow discontinuity and tiny hole in flat region. Fulfill the disconnected boundary.

The following figure is the result of erode and dilate operations. The first image is the original image. The second image is the result of erode operation. The third image is the result of dilate operation.

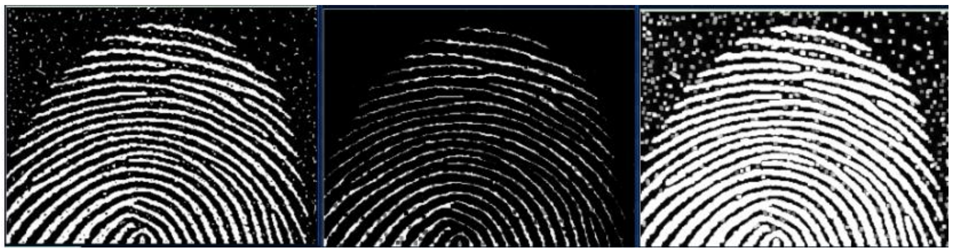


Figure 2 : erode and dilate result

The combination of open and close can make the image boundary smooth.

Partial important functions:

```
1.cvErode( const CvArr* src, CvArr* dst,
    IplConvKernel* element CV_DEFAULT(NULL),
    int iterations CV_DEFAULT(1)  );
```

Src is input image and dst is the output result after eroding. The third parameter is the shape o f window. The default value of it is a 3*3 rectangle window. The last window is how many times of iteration.

2. cvDilate(const CvArr* src, CvArr* dst,

IplConvKernel* element CV_DEFAULT(NULL),

int iterations CV_DEFAULT(1));

Src is input image and dst is the output result after dilating. The third parameter is the shape of window. The default value of it is a 3*3 rectangle window. The last window is how many times of iteration.

(3) cvCreateStructuringElementEx(int cols, int rows, int anchor_x, int anchor_y, int shape, int* values CV_DEFAULT(NULL));

The first two parameters are the size of window. The third and fourth parameters are the cen ter(or say anchor) you want to define in the window. The fifth parameter is the type of wind ow(it can be rectangle, cross, oval or even customer DIY window). The last parameter is the window you want to design if you choose a customer DIY window in fifth parameter.

2.2.2 largest connected domain

Since we have split noise in the threshold image, we need to find the largest connected domain to focus on and capture the Region of Interests(ROI). It is a smart way to ignore the useless region and get rid of noise. A example is showed as follow,

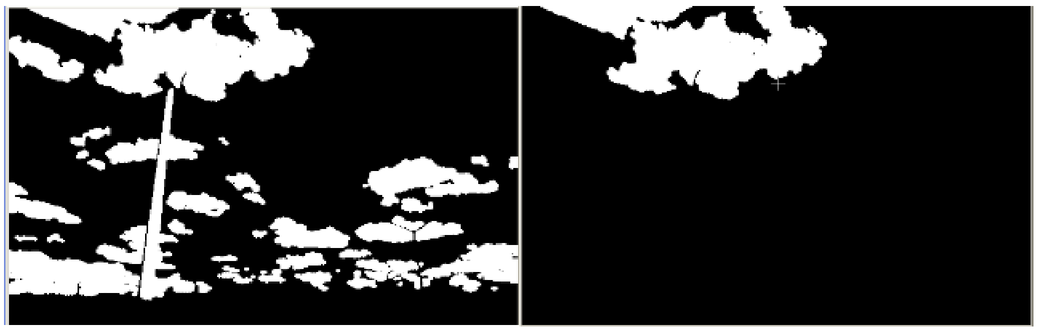


Figure 3: Threshold image(left) ; Results after finding largest connected domain

However, in this implementation, we didn't calculate the largest connected domain because the region of hand in image might have slight crack. All region for hand should be kept so that a threshold is set to check the size of each connected domain. If the size of a connected domain is greater than threshold, it will be kept as useful information. Otherwise, it will be filtered as noise.

Partial important functions:

int cvFindContours(CvArr* image, CvMemStorage* storage, CvSeq** first_contour,int header_size=sizeof(CvContour), int mode=CV_RETR_LIST,int method=CV_CHAIN_APPROX_SIMPLE, CvPoint offset=cvPoint(0,0));

Image: binary source input image

Storage: Save the contours

header_size: the length of header sequence.For example, if method=CV_CHAIN_CODE, header_size=sizeof(cvChain).

Mode:searching mode (options:CV_RETR_EXTERNAL,CV_RETR_LIST,

CV_RETR_CCOMP,CV_RETR_TREE)

Method:edge approximation methods

Offset: bias of contour

2.3 Module-3

Just like previous two modules, even this module uses OpenCV. The most important OpenCV functions used here are

1. Contrast Limited Adaptive Histogram Equalization(CLAHE).
2. Harrcascade Classifier.
3. Adaptive Threshold.

2.3.1 Contrast Limited Adaptive Histogram Equalization

Definition:

Adaptive histogram equalization (AHE) is an image processing method that is used to enhance contrast in the input images. It is different from ordinary histogram equalization in the respect that the adaptive method computes several histograms for different sections of an image and based on that redistribute the contrast values in the image. It is therefore suitable for improving the local contrast and enhancing the definitions of edges in each region of an image.

Contrast Limited AHE (CLAHE) differs from general adaptive histogram equalization in its contrast limiting. This feature can also be applied to global histogram equalization, giving rise to contrast limited histogram equalization (CLHE), which is rarely used in practice. In the case of CLAHE, the contrast limiting procedure has to be applied for each neighborhood from which a transformation function is derived. Figure-1 describes how the contrast limiting is done.

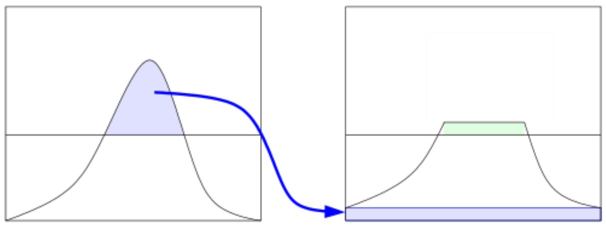


Figure 4 : CLAHE

Function declaration:
clahe = cv2.createCLAHE(clipLimit, tileGridSize)

Parameters:

clipLimit	– Normalized contrast limit, above which the pixels are going to be
tileGridSize	-- Window size

Return:

Clache	-- Image returned after CLAHE.

2.3.2 Harr Cascade Classifier

Definition:
Object Detection by Haar feature-based cascade classifiers is a better way of object detection method presented by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features". They are just like the convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under white rectangle from sum of pixels under black rectangle.

The functions extract features from the image and then matches them with the features present in the xml file and this is not one -many matching. They are cascaded to make the entire process faster. For example, 24x24 window has 160000 features, so looking for a match in every region is exhaustive process, so initially all the negative detections are removed and then features to be matched against cascaded.

Function declaration:
cv.HaarDetectObjects(image, cascade, storage, scale_factor, min_neighbors, flags, min_size,))

Parameters:
1.cascade –It can be loaded from XML or YAML file using Load(). When the cascade is not needed anymore.
2.image – Matrix of the type CV_8U containing an image where objects are detected.
3.objects – Vector of rectangles where each rectangle contains the detected object.
4.scaleFactor – Parameter specifying how much the image size is reduced at each image scale.
5.minNeighbors – Parameter specifying how many neighbors each candidate rectangle should have to retain it.
6.flags – Parameter with the same meaning for an old cascade as in the function cvHaarDetectObjects. It is not used for a new cascade.
7.minSize – Minimum possible object size. Objects smaller than that are ignored.
8.maxSize – Maximum possible object size. Objects larger than that are ignored

2.3.2 Adaptive Threshold

Definition:
In the general thresholding, we used a global value as threshold value. This value may not be good where image has different lighting conditions in different regions of image. In this case, the we calculate the threshold for a small regions of the image. We get various thresholds for different regions of the same image and it gives us good results for images with varying illumination.

Function declaration:
cv.AdaptiveThreshold(src,dst,maxValue,adaptive_method=CV_ADAPTIVE_THRESH_MEAN_C, thresholdType=CV_THRESH_BINARY, blockSize=3, param1=5)

Parameters:

1.src – Source 8-bit single-channel image.

2.dst – Destination image of the same size and the same type as src .

3.maxValue – Non-zero value assigned to the pixels for which the condition is satisfied. See the details below.

4.adaptiveMethod – Adaptive thresholding algorithm to use, ADAPTIVE_THRESH_MEAN_C or ADAPTIVE_THRESH_GAUSSIAN_C . See the details below.

5.thresholdType – Thresholding type that must be either THRESH_BINARY or THRESH_BINARY_INV .

6.blockSize – Size of a pixel neighborhood that is used to calculate a threshold value for the pixel: 3, 5, 7, and so on.

CHAPTER 3
APPROACH

3.1 Color recognition & Tracking

3.1.1 Tool used:

Programming language: C++

Library used: openCV 2.4.8

Operating system: Ubuntu 15.04

3.1.2 Flow chart

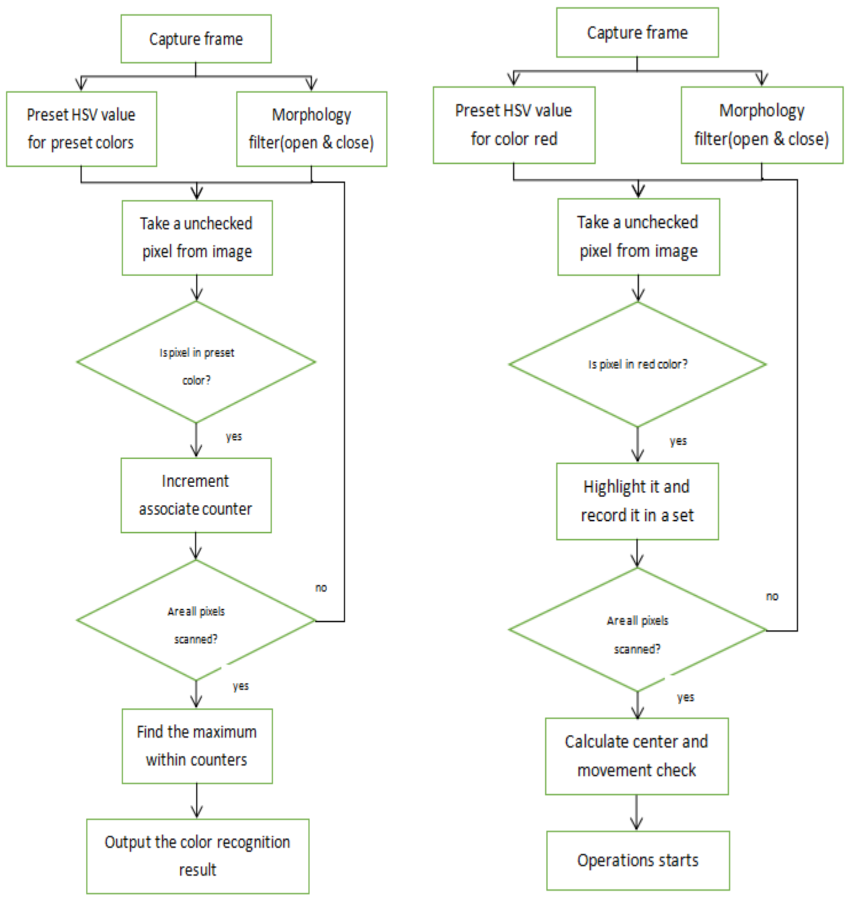

Capture frame
Preset HSV value for preset colors
Morphology filter(open & close)
Take a unchecked pixel from image
Is pixel in preset color?
yes
Increment associate counter
Are all pixels scanned?
no
yes
Find the maximum within counters
Output the color recognition result
Capture frame
Preset HSV value for color red
Morphology filter(open & close)
Take a unchecked pixel from image
Is pixel in red color?
yes
Highlight it and record it in a set
Are all pixels scanned?
no
yes
Calculate center and movement check
Operations starts

Figure 5 : flow chart of color recognition(left) ; flow chart of color tracking(right)

3.1.3 Description of color recognition

(1) The flow chart in the left is for color recognition part. First, we need to use OpenCV function cvCapture() to capture the camera frame. Here we set the output frame is a 640*480 pixels image. Then the HSV threshold value for each preset color will be set as Table 1 in section 2.1. A morphology filter will also help us to smooth the boundaries in image. After that, we start a scanning for all pixels in image. If a pixels meets the requirements of all HSV threshold, it will be considered as a particular preset color and the counter of associate color will be added by 1. Otherwise, a pixel will be ignored. After we scanned and judge all the pixels in image, we can find which color is dominated in the image. Since the back ground color is skipped and skin color would be divided, the main color in the figure should be your shirt color.

(2) The flow chart in the right is for color tracking part. The first several steps are the same as color recognition part. First, we need to capture the camera frame. In this particular part, we are only interested in red color so that the HSV threshold of red color will be set. Then, open operation and close operation are both used to smooth the edges in the image.After that, we will also start a scanning for all the pixels in one frame. If a certain pixel is in red color, highlight it and record it in a set for further calculation. Otherwise, a pixel will be ignored. When all the pixels are scanned, we can calculate the center of target color region. Compared the location of current center to previous center, if the difference exceed a threshold, , a movement direction will be detected and the corresponding operating will start.

3.2 Hand & Gesture recognition

3.2.1 Tool used:

Programming language: C++

Library used: OpenCV 2.4.8

Operating system: Ubuntu 15.04

3.2.2 Flow chart

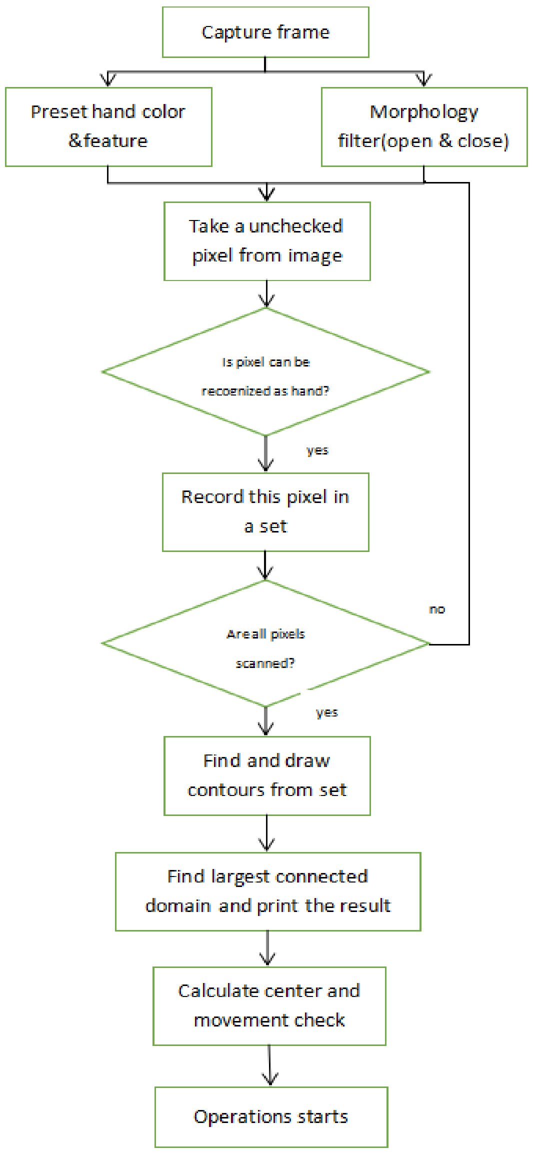

Capture frame
Preset hand color &feature
Morphology filter(open & close)
Take a unchecked pixel from image
Is pixel can be recognized as hand?
yes
Record this pixel in a set
Are all pixels scanned?
no
yes
Find and draw contours from set
Find largest connected domain and print the result
Calculate center and movement check
Operations starts

Figure 6 : flow chart of hand & gesture recognition

3.2.3 Description of flow chart

First, we need to capture the camera frame. Then the hand feature will be set in the following two steps.

(1) Cut the whole image into several blocks by the edge and boundary

(2) Extract the color which is most similar to human skin.

In this setting, we are seeking pixel with following HSV value:

Section	H	S	V
1	0-14	95-100	-
2	0-30	20-100	-
3	170-180	15-90	-

Table 2 : Human skin color threshold

Any pixel meets the requirement of all these 3 threshold will be considered as a potential hand pixel. Besides the color requirement, the size of connected domain will also be checked. That is to say, if there is a large size of connected pixels which are in the interval of HSV threshold, they will be regarded as hand. Otherwise, a single pixel which is in the interval of HSV threshold will also be ignored.

Then, morphology filter will be implemented to help us remove the noise. In this filter, we need both open operation to recover the crack and close operation to remove the holes in flat area. After that, we will start to scan all the pixels on one frame. All connect domains will be printed in the window.

When we got all connect domains, we will filter those domains again by setting a threshold for domain size. In other words, small connect domains will be considered as noise and will be deleted. The remain domains should be the qualified domains.

Finally, we can compute the center of qualified domains. Compared the current center with the center of last frame, if the difference exceed a threshold, we can assume there is a movement happen.

3.3. Smile detection

3.3.1 Tools used

Machine	Lenovo Think Pad- E550 i7 5th gen
Operating system	Ubuntu 14.0.4

Software tools/libraries:

1.OpenCV – version 2.4.8

2.Python -- version 2.7

Camera :

Logitech – C200

3.3.2 Flow Chart

3.3.2.1 Flow Chart for static data set

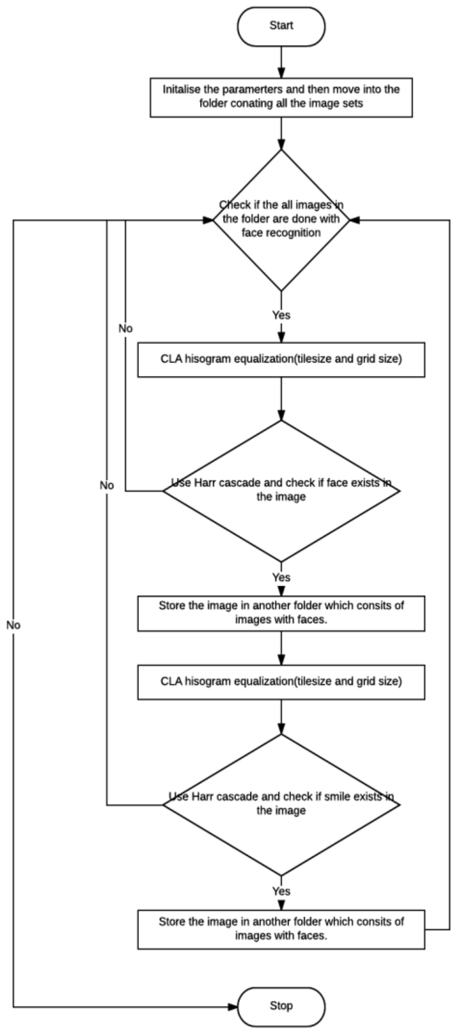

Start
Initalise the paramerters and then move into the folder conating all the image sets
Check if the all images in the folder are done with face recognition
Yes
No
CLA hisogram equalization(tilesize and grid size)
Use Harr cascade and check if face exists in the image
No
Yes
Store the image in another folder which consits of images with faces.
No
CLA hisogram equalization(tilesize and grid size)
Use Harr cascade and check if smile exists in the image
Yes
Store the image in another folder which consits of images with faces.
Stop

Figure 7 : Flow chart- Smile detection – static data set

3.3.2.1 Flow Chart for real time data.

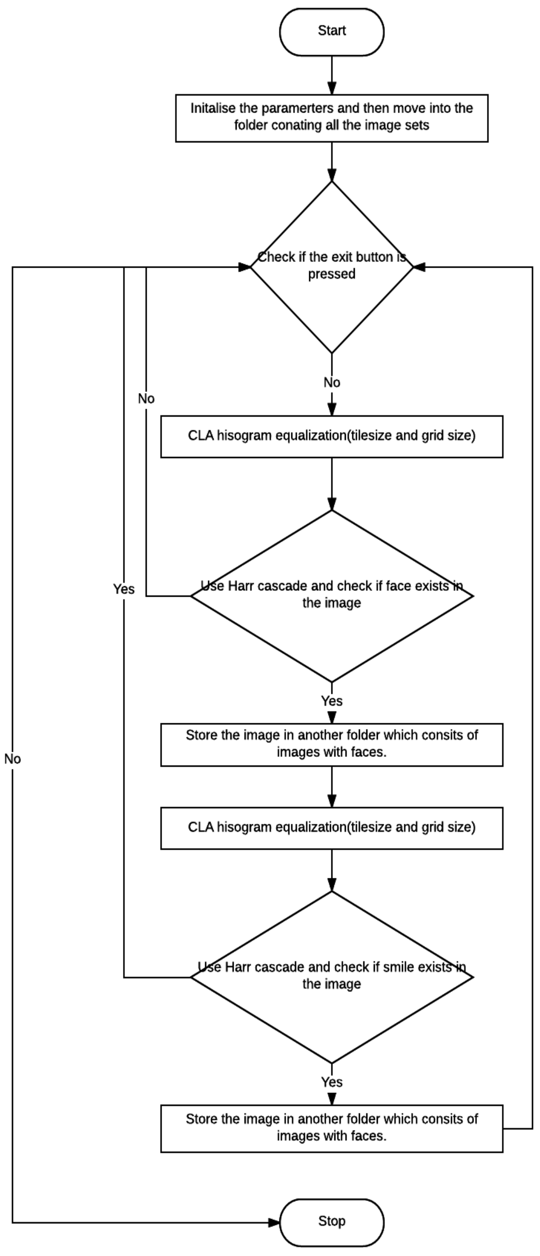

Start
Initalise the paramerters and then move into the folder conating all the image sets
Check if the exit button is pressed
No
CLA hisogram equalization(tilesize and grid size)
No
Use Harr cascade and check if face exists in the image
Yes
Yes
Store the image in another folder which consits of images with faces.
CLA hisogram equalization(tilesize and grid size)
No
Use Harr cascade and check if smile exists in the image
Yes
Store the image in another folder which consits of images with faces.
Stop

Figure 8 : Flow chart- Smile detection – real time data set

3.3.3 Description of the flow chart

3.3.3.1 Description of the flow chart for static data set

Initialize the parameters for Harr Cascade to adjust the maximum and minimum limits of the size and also scale factor. Using os.chdir in python move into the folder where all the images are stored. At the stage start taking photos one by one in the all_photo directory, then apply CLAHE to reduce the noise. Initialize the grid size and clip limit for CLAHE. Use Harr Cascade to detect presence of the face and if the face is present then save the image with the blue border around the face. Next use the harr Cascade to detect the presence of smile in the lower half of the detected face and if there is a smile then save the image with green border. Now detect the eyes in the upper half of the face detected. At the

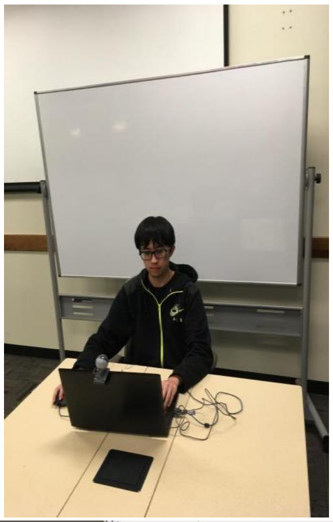
3.3.3.2 Description of the flow chart for real time data set

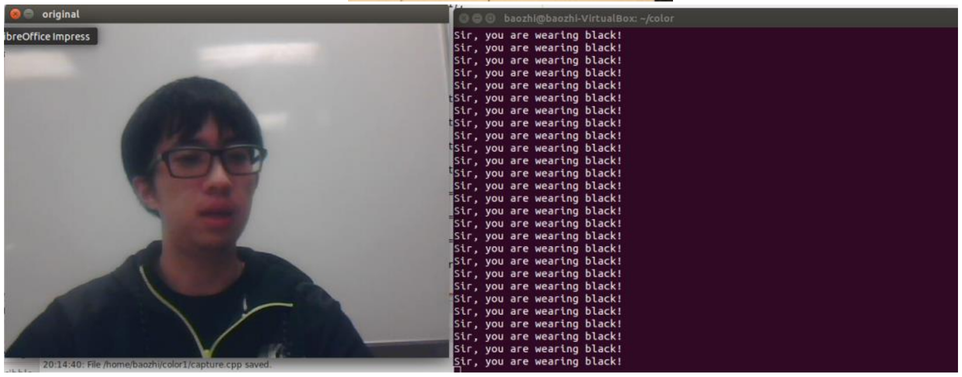
The main part that is different from 3.3.3.1 is the image is a real time image taken from a camera C200 and displaying the image in the screen and also saving the video.

CHAPTER 4
DATA ANALYSIS AND RESULTS

4.1 Color recognition & Tracking

(1) Black wearing trial





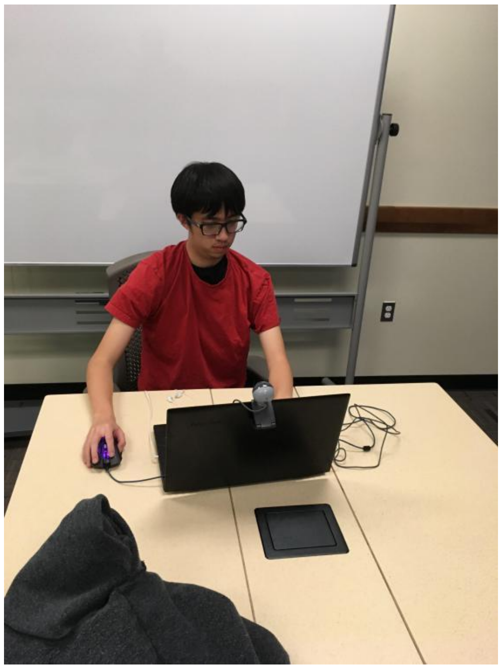
```
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
Sir, you are wearing black!
```

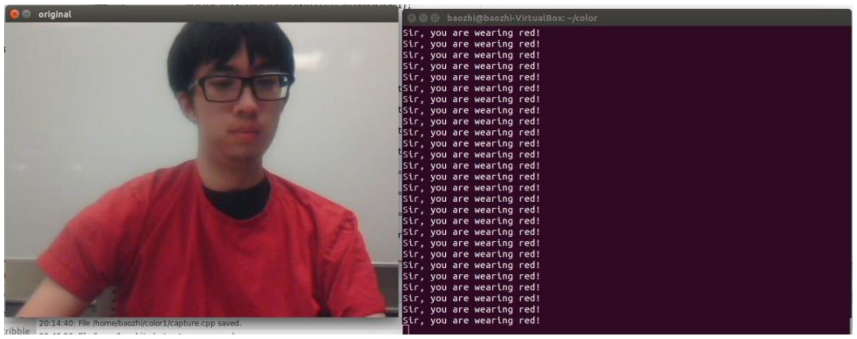
```
20:14:40: File /home/baozhi/color1/capture.cpp saved.
```

Figure 9,10: Black wearing setting and result

In this part, I wear a black sweater and my partner take the photo for me. From the terminal, we can see the result "Sir, you are wearing black".

(2) Red wearing trial





```
baozhi@baozhi-VirtualBox: ~/color
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
Sir, you are wearing red!
```

Figure 10,11: Red wearing setting and result

In this part, I wear a red shirt and my partner take the photo for me. From the terminal, we can see the result “Sir, you are wearing red”.

(3) Red tracking trials

In this part, we define 4 operations.

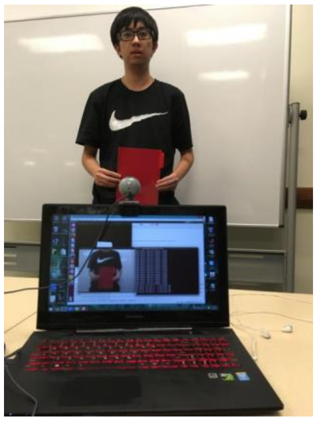
a. If the red object is moving down, the result “Down option: The alarm is : 11:30 pm Dec 4th” is showed in terminal.

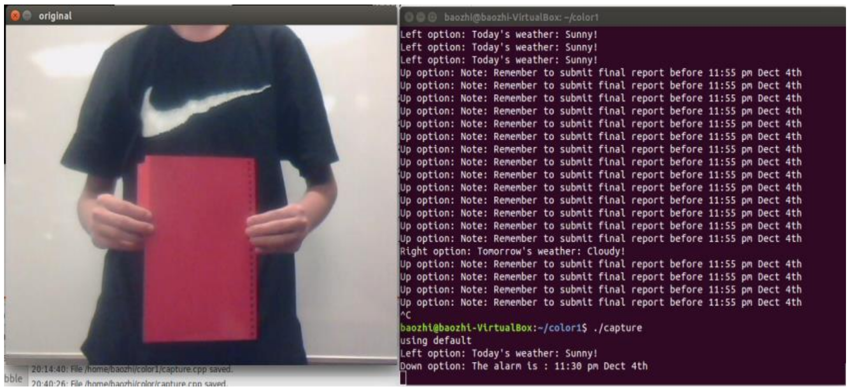
b. If the red object is moving down, the result “Up option: Note: Remember to submit report before 11:55 pm Dec 4th” is showed in terminal.

c. If the red object is moving to left, the result “Left option: Note: Today’s weather : sunny!” is showed in terminal.

d. If the red object is moving to right, the result “Right option: Tomorrow’s weather : Cloudy!” is showed in terminal

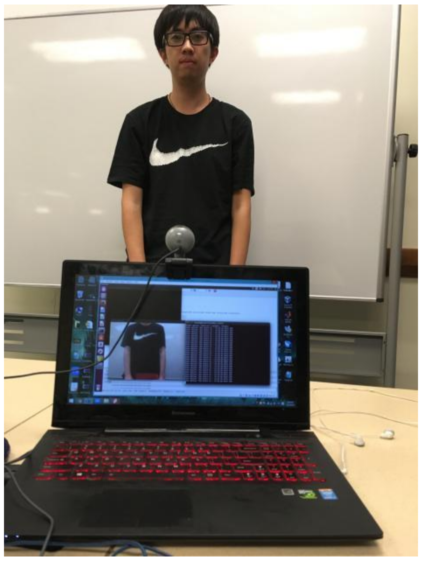
There are 5 pairs of figure will be showed as follow,

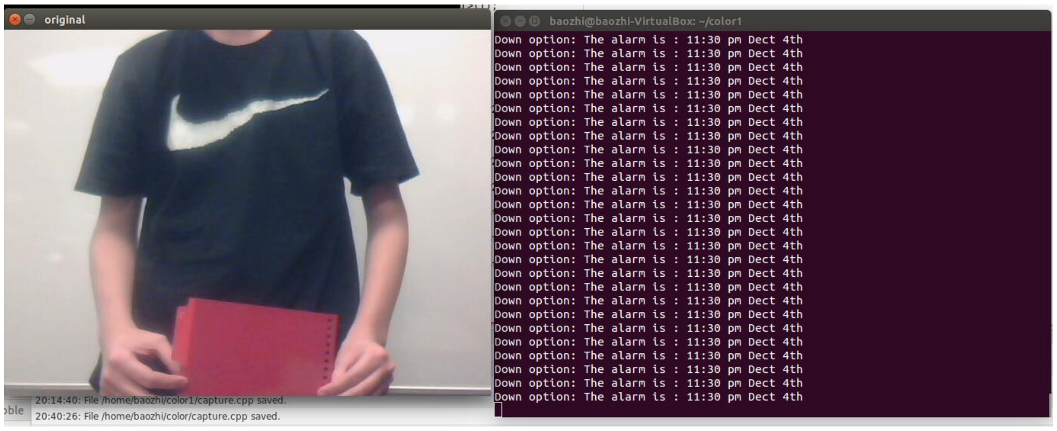



original
baozhi@baozhi-VirtualBox: ~/color1
20:14:40: File /home/baozhi/color1/capture.cpp saved.
20:40:26: File /home/baozhi/color/capture.cpp saved.

```
Left option: Today's weather: Sunny!
Left option: Today's weather: Sunny!
Left option: Today's weather: Sunny!
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Right option: Tomorrow's weather: Cloudy!
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
^C
baozhi@baozhi-VirtualBox:~/color1$ ./capture
using default
Left option: Today's weather: Sunny!
Down option: The alarm is : 11:30 pm Dect 4th
```

Figure 11,12: Initial setting and result

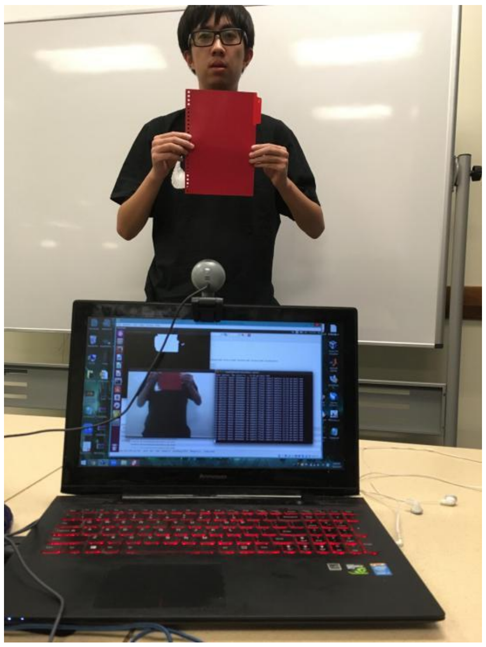


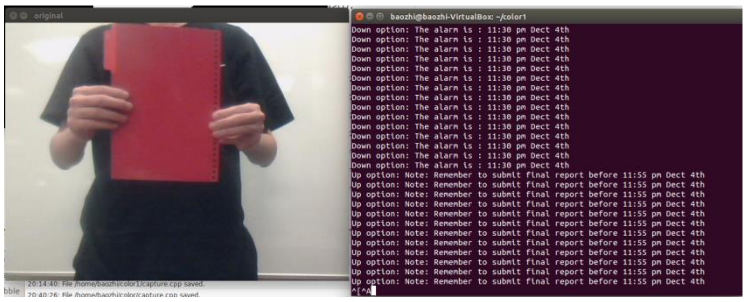


```
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
```

```
20:14:40: File /home/baozhi/color1/capture.cpp saved.
20:40:26: File /home/baozhi/color/capture.cpp saved.
```

Figure 13,14: Downward movement and result

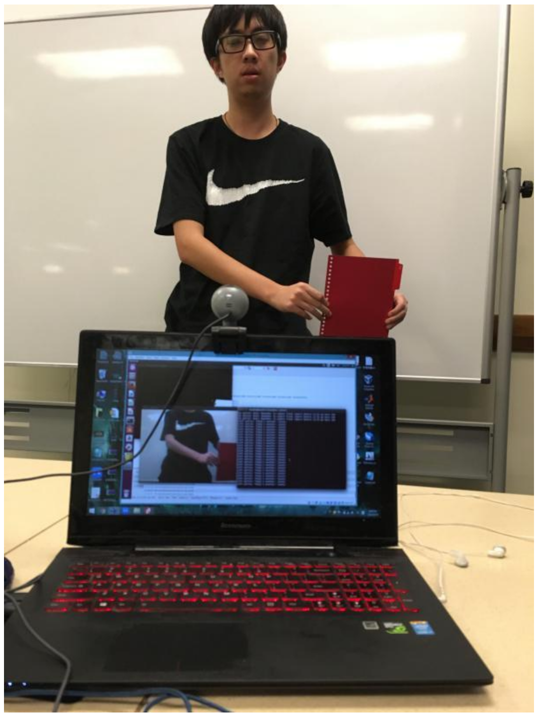


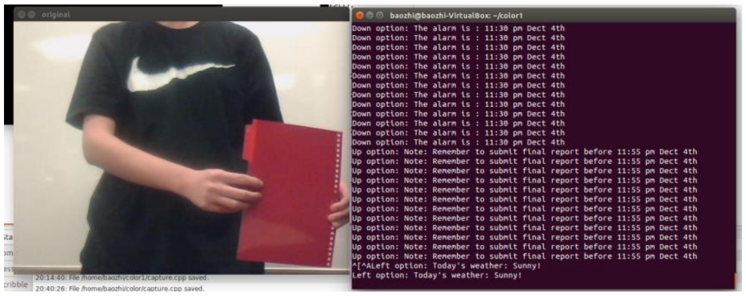


```
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
^[^A
```

```
20:14:40: File /home/baozhi/color1/capture.cpp saved.
20:40:26: File /home/baozhi/color/capture.cpp saved.
```

Figure 15,16: upward movement and result

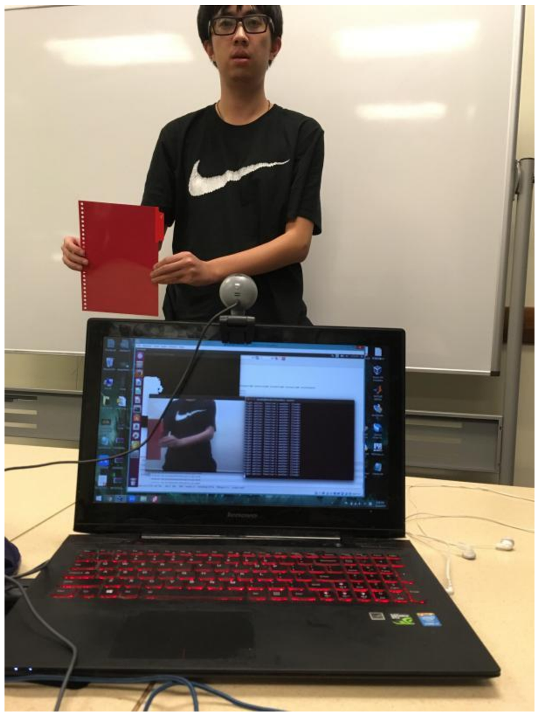


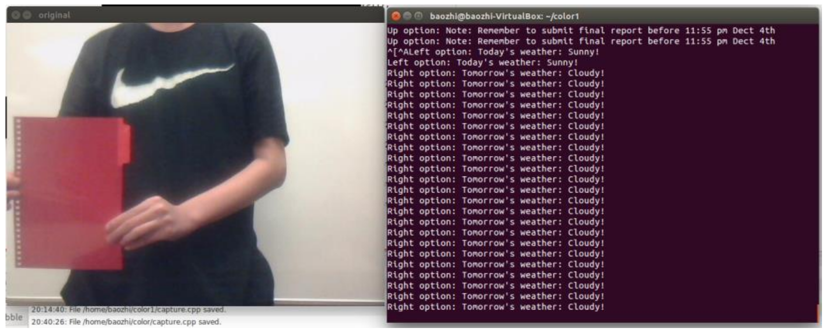


```
20:14:40: File /home/baozhi/color1/capture.cpp saved.
20:40:26: File /home/baozhi/color/capture.cpp saved.
```

```
baozhi@baozhi-VirtualBox: ~/color1
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
^[^ALeft option: Today's weather: Sunny!
Left option: Today's weather: Sunny!
```

Figure 17,18: left movement and result





```
20:14:40: File /home/baozhi/color1/capture.cpp saved.
20:40:26: File /home/baozhi/color/capture.cpp saved.
```

```
baozhi@baozhi-VirtualBox: ~/color1
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
^[^ALeft option: Today's weather: Sunny!
Left option: Today's weather: Sunny!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
```

Figure 19,20: right movement and result

Analysis: With the result above, we verify that the color recognition and color tracking triggered operations are working well. Since it's a real time and real world task, it is more important to make it work in real time. We also set a timing profiler to record the timing spent on each frame.

For color recognition part: the fps is about 14 which means it will take 0.07s to process each frame.

For color tracking part: the fps is about 11 which means it will take 0.09s to process each part.

Limitations:

(1) Background should be simple(white preferred) and stable. If not, the threshold image will be noisy and the output result will be inaccurate.

(2) light should be fit. If the light is too bright, all the pixels will be considered as white pixels or lighter than it should be.

4.2 Hand & Gesture recognition

In this section, several figures will be showed to demonstrate hand & gesture recognition is done.

In this section, we also define 4 operations.

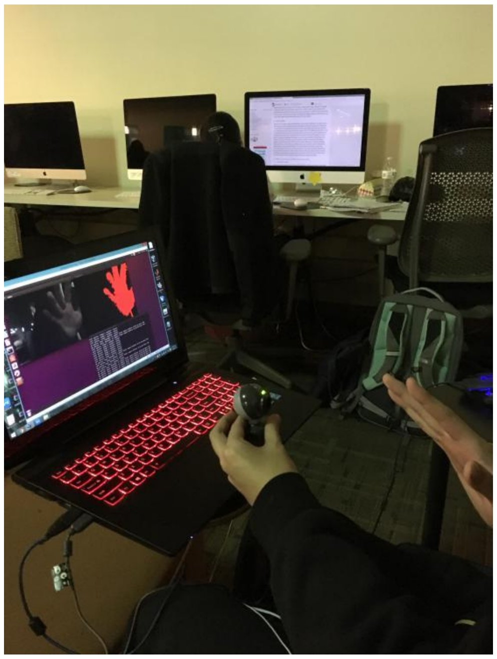
a. If the hand is moving down, the result “Down option: The alarm is : 11:30 pm Dec 4th” is showed in terminal.

b. If the hand is moving down, the result “Up option: Note: Remember to submit report before 11:55 pm Dec 4th” is showed in terminal.

c. If the hand is moving to left, the result “Left option: Note: Today’s weather : sunny!” is showed in terminal.

d. If the hand is moving to right, the result “Right option: Tomorrow’s weather : Cloudy!” is showed in terminal.

The images are as follow,



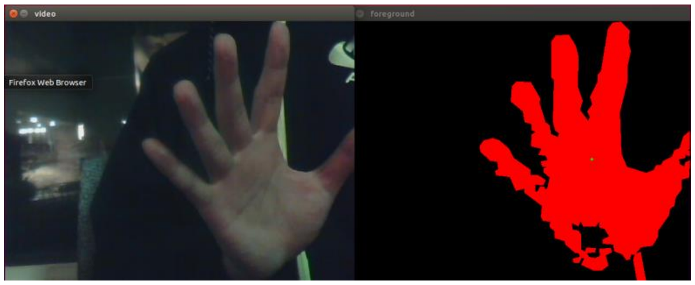
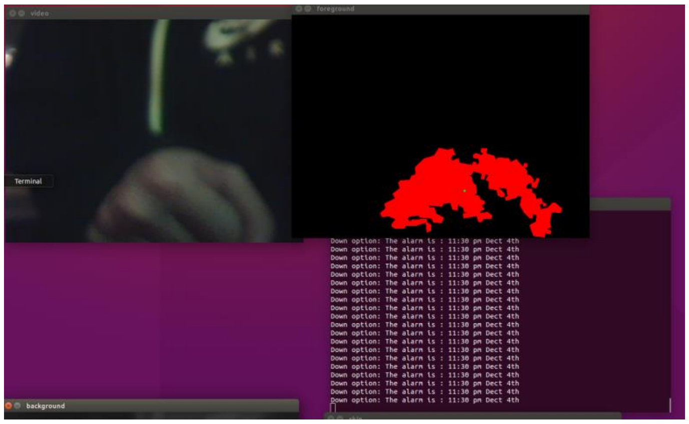
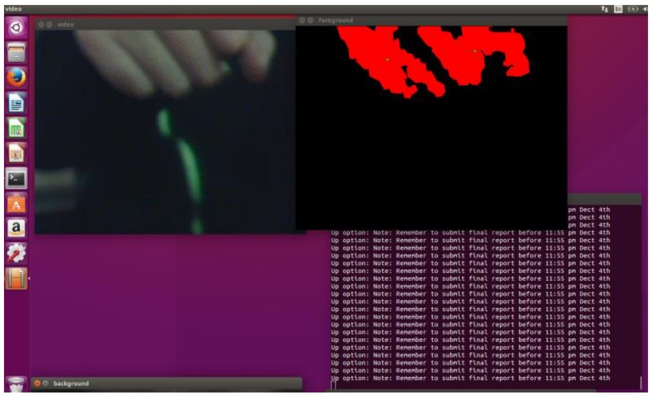


Figure 21,22: Hand recognition setting and result



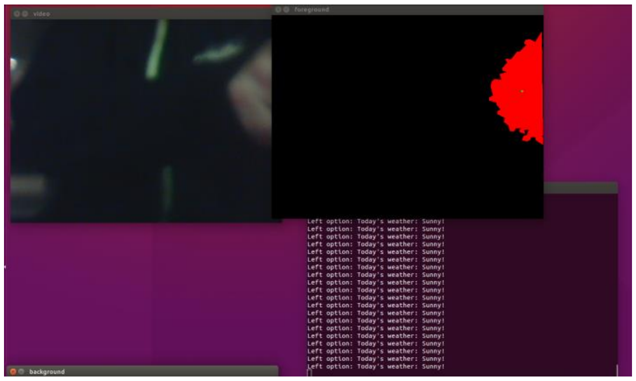
```
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
Down option: The alarm is : 11:30 pm Dect 4th
```

Figure 23: downside movement result



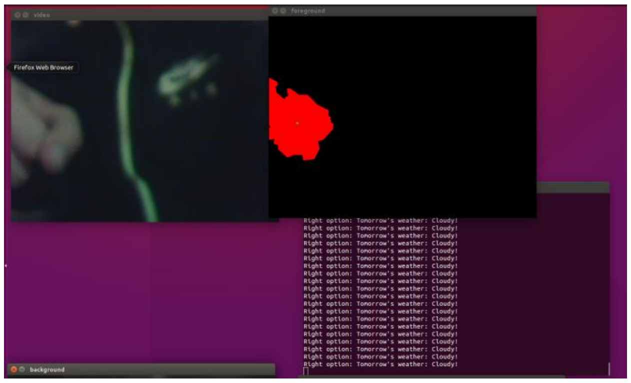
```
                                                  pm Dect 4th
                                                  pm Dect 4th
                                                  pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
Up option: Note: Remember to submit final report before 11:55 pm Dect 4th
```

Figure 24: Upward movement result



```
Left option: Today's weather: Sunny!
Left option: Today's weather: Sunny!
Left option: Today's weather: Sunny!
Left option: Today's weather: Sunny!
Left option: Today's weather: Sunny!
Left option: Today's weather: Sunny!
Left option: Today's weather: Sunny!
Left option: Today's weather: Sunny!
Left option: Today's weather: Sunny!
Left option: Today's weather: Sunny!
Left option: Today's weather: Sunny!
Left option: Today's weather: Sunny!
Left option: Today's weather: Sunny!
Left option: Today's weather: Sunny!
Left option: Today's weather: Sunny!
Left option: Today's weather: Sunny!
Left option: Today's weather: Sunny!
Left option: Today's weather: Sunny!
Left option: Today's weather: Sunny!
Left option: Today's weather: Sunny!
```

Figure 25: Left movement result



```
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
Right option: Tomorrow's weather: Cloudy!
```

Figure 26: Right movement result

Analysis: From the results above, we verify that we can recognize human hand and human skin. Also, 4 different operations are triggered when hand waves to different direction. For timing analysis, it is slower than color recognition part. The fps of hand recognition is only about 5 which means it take 0.2s to process each frame.

Limitations:

(1) Background should be simple(black preferred) and stable. The reason why we need a black background is that the color of my hand is light so that a black background will offer a strong contrast.
(2) light should be fit.

Also, we try to recognize paper, scissors and rocks by human hands and play games with computer. However, by the final project report due time, we didn't finish it yet. We will keep trying it later and, if possible, some result will be showed in presentation.

4.3 Smile detection

4.3.1 Smile detection using static data set

Data set:

This data set is taken from Yale university website and specifically the images are from the dataset of 96 and 97. These images are taken for face recognition but not specifically for smile

Total number of people	52
Number of men	25
Number of women	27
Number of photos per each person	16
Total number of images	7432

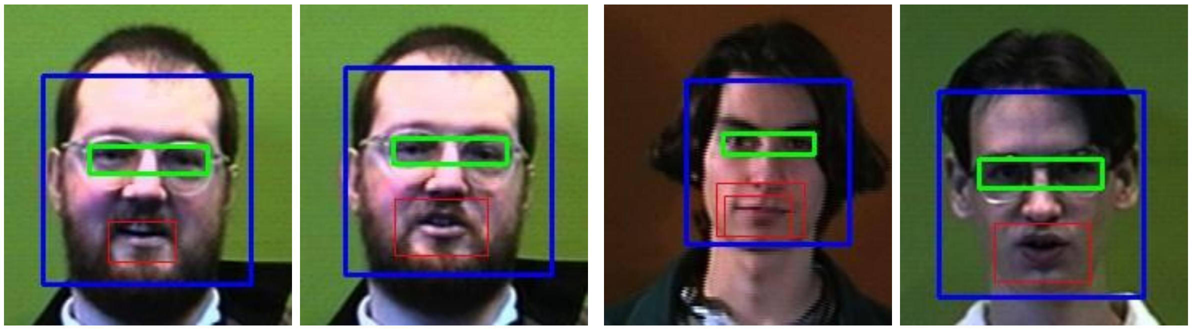
Initialization parameters

1.CLAHE – clipLimit , Grid size

2.HarrCascade -- Scale factor, Minimum neighbors

Total number of images

Clip limit	Grid size	Scale factor	Minimum neighbour	Number of images Where faces detected	Number of images--smiles	Number of images--eyes	Face Correct Prediction %	Smile correct predictio n %	Eye Correct Prediction %
2	5,5	1.1	5	7060	237	5238	0.96	0.63	0.86
1.9	7,7	1.3	6	7048	61	5133	0.95	0.35	0.83
1.8	9,9	1.5	7	7044	40	5055	0.94	0.32	0.80
1.7	2,2	1.7	8	7123	19	5377	0.97	0.40	0.87
1.6	3,3	1.1	5	7117	125	5239	0.97	0.30	0.85



After face , eye and smile detection

Limitations:

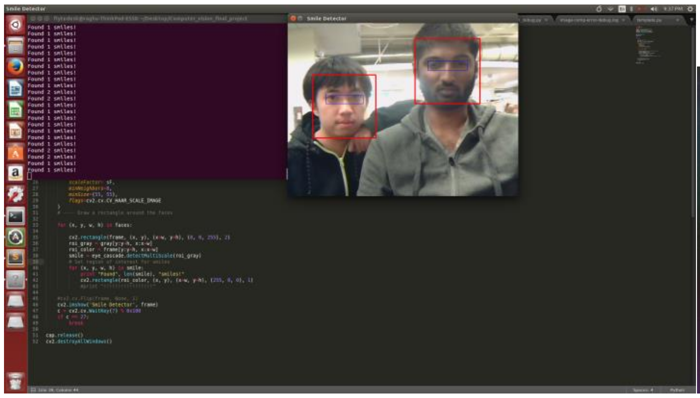
1. Unequal distribution of light even after CLAHE
2. Cannot identify images eyes when spectacles in low light conditions
3. Very bad training set for smile detection
4. When the head is rotated there is good possibility that face cannot be detected

Dataset: https://drive.google.com/open?id=0B965WtzIU_oqaDJqcFRubHFITmc

4.3.2 Smile detection using real time data set

Here the results are videos that are placed in the results folder.

Certain screen shots of the video are place here and also the test setup image are pasted below.



```
Found 1 smiles!
Found 1 smiles!
Found 1 smiles!
Found 1 smiles!
Found 1 smiles!
Found 1 smiles!
Found 1 smiles!
Found 1 smiles!
Found 1 smiles!
Found 1 smiles!
Found 2 smiles!
Found 2 smiles!
Found 1 smiles!
Found 1 smiles!
Found 1 smiles!
Found 1 smiles!
Found 1 smiles!
Found 1 smiles!
Found 2 smiles!
Found 2 smiles!
Found 1 smiles!
Found 1 smiles!
```

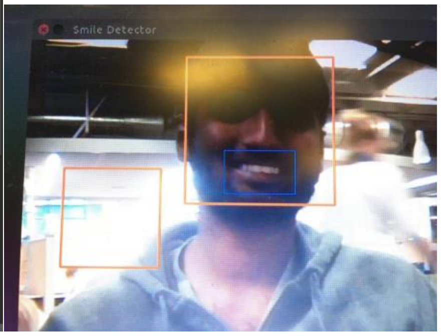
```
            scaleFactor= sF,
            minNeighbors=8,
            minSize=(55, 55),
            flags=cv2.cv.CV_HAAR_SCALE_IMAGE
        )
        # ---- Draw a rectangle around the faces

        for (x, y, w, h) in faces:

            cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255), 2)
            roi_gray = gray[y:y+h, x:x+w]
            roi_color = frame[y:y+h, x:x+w]
            smile = eye_cascade.detectMultiScale(roi_gray)
            # Set region of interest for smiles
            for (x, y, w, h) in smile:
                print "Found", len(smile), "smiles!"
                cv2.rectangle(roi_color, (x, y), (x+w, y+h), (255, 0, 0), 1)
                #print "!!!!!!!!!!!!!!!!!!!!"

        #cv2.cv.Flip(frame, None, 1)
        cv2.imshow('Smile Detector', frame)
        c = cv2.cv.WaitKey(7) % 0x100
        if c == 27:
            break

cap.release()
cv2.destroyAllWindows()
```



CHAPTER 4
CONCLUSION

4.1 Lessons learned

1.Light is the main factor in all the three modules, so understanding your surrounding basically all light sources are important. The results we got in the night tube light as the main source of light is different from result in morning where the main source of light is sun.
2.Harr Cascade though considered to be the one of the best face recognition technique in OpenCV has many limitations either due to the training set or due to the number of images in the training set, though 6000 appears to be a large number for a training set but with various conditions of illumination, they fall short in number.
3.Before planning any real time image processing testing, a test set up should be the thing in your mind.
4.In color recognition and tracking part, we learned a new color model HSV. Compared to RGB color model, it is a better to describe the format of a color.
5. During the work of hand & gesture recognition, we know the importance of contrast.

4.2 Future scope

1. When we implemented color recognition work, we also tried to use largest connected domain method to find the main color. But there might be folding on it so that the cloth will be cut. More work should be done to solve this problem.
2. During the work of hand & gesture recognition, we tried to implement paper, scissors and rock recognition. With the limitation of time, we didn't finish this part by the due of this report.

4,3 Advice to future students

Make sure that taking care of surroundings like light and contrast when you are trying to implement Computer Vision related topic in real time.

For homework, try to avoid using nested for loops because it is so time taking.

4,4 Conclusion

From this project, we have a better understanding about recognition work. For color recognition and tracking part, we extract target color and identify color by HSV color model. The work of hand & gesture recognition inspired us about morphology filter and its operations. Also, we learned to use largest connect domain method to find the Region of Interest (ROI). Smile detection using Harr Cascade is said to best among the OpenCV but once we went through them we can across various flaws when tested with different conditions.

REFERRENCES

[1] http://blog.csdn.net/augusdi/article/details/9009259

[2] http://blog.csdn.net/lg1259156776/article/details/50085045

[3] http://blog.csdn.net/liuqz2009/article/details/47623399

[4] http://www.ti.com/lit/wp/spry199/spry199.pdf

[5] http://opencv.org/

[6] http://docs.opencv.org/3.0-beta/modules/face/doc/facerec/index.html

[7] http://docs.opencv.org/2.4/modules/contrib/doc/facerec/tutorial/facerec_video_recognition.html

[8]http://aimm02.cse.ttu.edu.tw/class_2009_2/CV/OpenCV/References/Orientation%20histograms%20for%20hand%20gesture.pdf

[9] http://dl.acm.org/inst_page.cfm?id=60021726&CFID=701936352&CFTOKEN=40645599

Google drive Link

Dataset: https://drive.google.com/open?id=0B965WtzIU_oqaDJqcFRubHFITmc