## Ideas

For my project I have decided to create a facial recognition system. This is to be used in conjunction with a camera attached to a raspberry pi which is positioned within the building in order to capture data. My aim is to display this data in a range of interesting and graphical ways.

## Languages

I intend to use Python to code the program for facial detection and recognition and the processing programming language to display the graphics for my project.

## Data capture

Ideas for the data I would like to capture include;

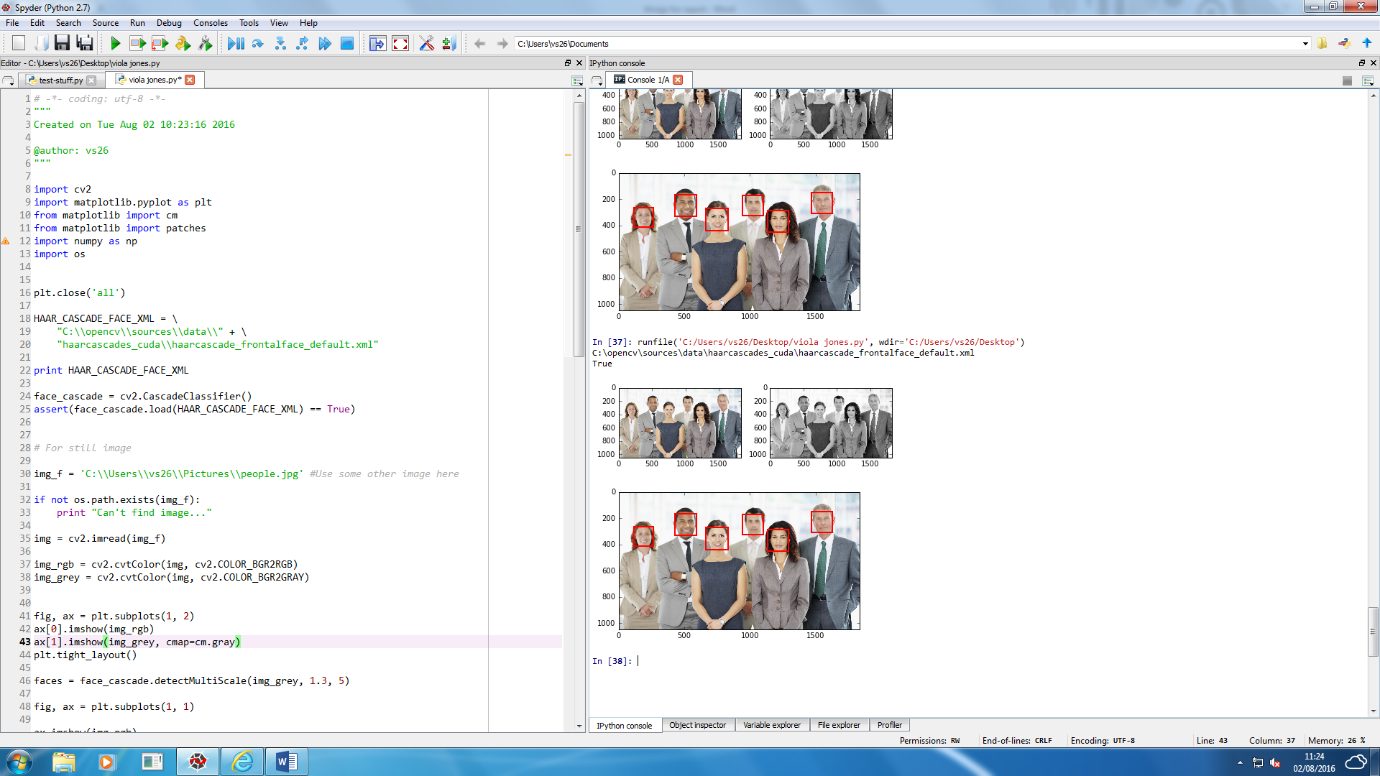
* How many people move past the camera per unit time
* How fast these people are moving
* Whether or not they stop
* Which direction they are moving
* How far away they are from the camera
* Groups? Average size?

## Displaying the data (ideas)

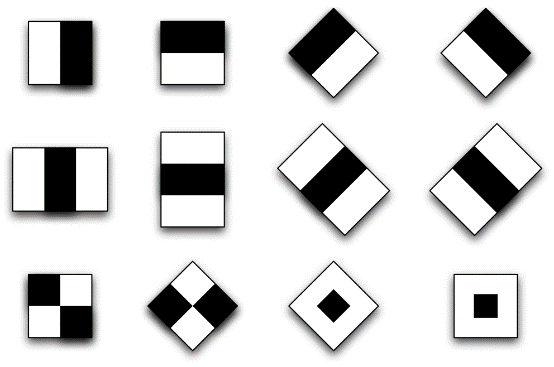
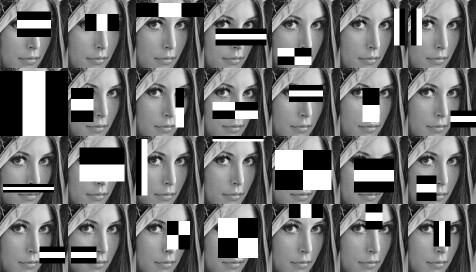
* Geometric shapes
  + Different opacity for how close they are to camera/how long stationary?
  + Change in colour/ colour of shape assigned due to (some parameter)
  + Rotation when head turns/change in direction i.e. rounding the corner on the stairs
  + Smooth movement across frame
  + Speed shape changes with the speed of movement
* Lines
  + A point represents the person
  + As the subject moves, leaves a trail
  + Trail may fade fast/slow depending on speed of movement
  + Change in colour/ colour of shape assigned due to (some parameter)
  + Thickness of trail may vary in respective to how close they are from the camera
  + ‘Points’/beams may pulse when stationary (?)

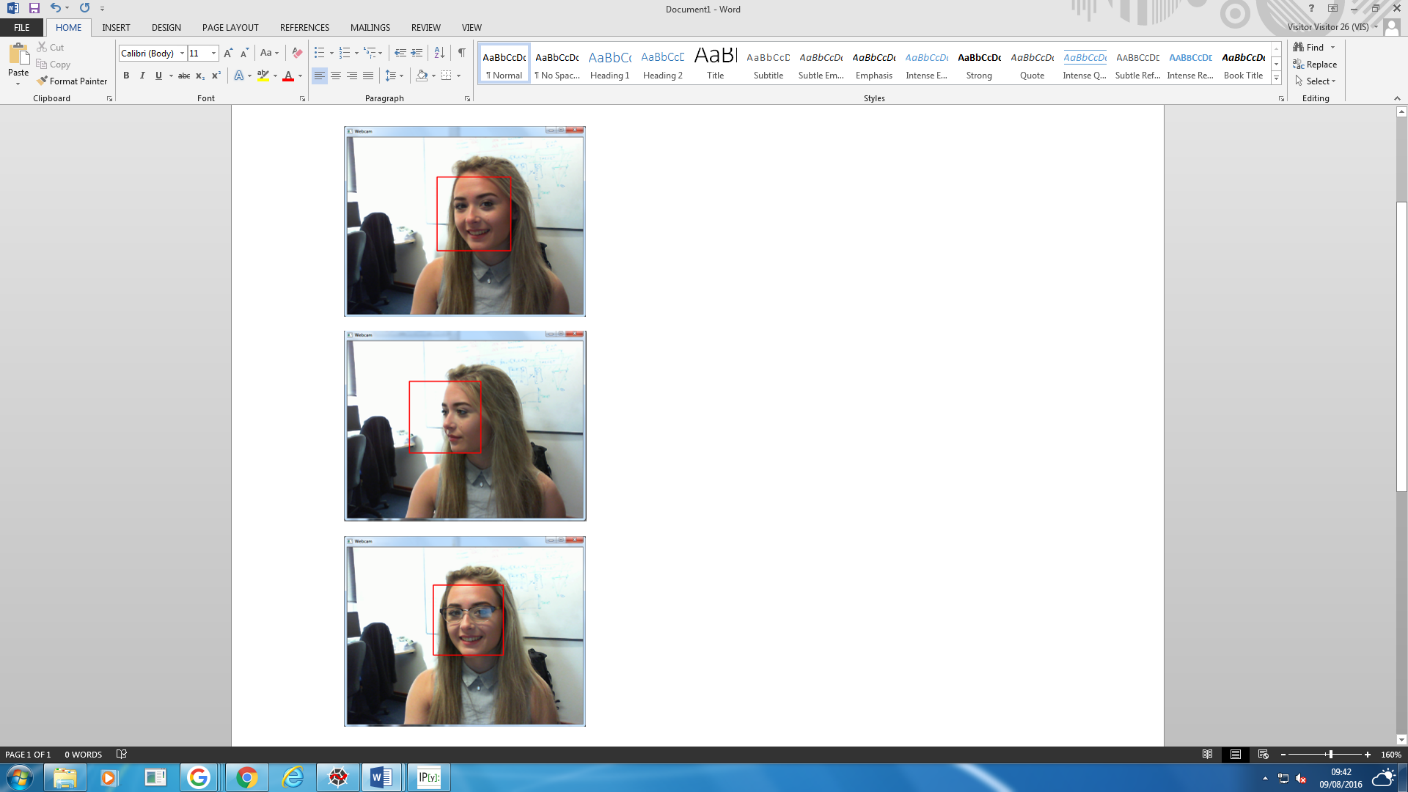
# ***Developments Log***

Firstly I needed to learn the basics of python, however I decided it may be best to learn as I went along (codcademy possibly?).

Primary task was to be able to detect a face in a frame. To do this, I read about multiple facial detection algorithms and decided that the Viola Jones algorithm would be the most effective for my project. I then took pieces of code offline and implemented them into a test program which ran facial detection on a set of still image, (see examples).

Viola Jones is a binary classifier that uses several weak detectors to verify if there is a face within an image; it does this by dividing the image into many overlapping rectangles which are then processed by a cascade, meaning each section of the image is compared to an object of instance which in this case are the Haar-like features that can be seen below. The black and white sections represent the intensity changes across an area.



The following task was being able to run this using a real-time frame rate. This involved looking at a few different parameters and using openCV for computer vision. (ADD MORE ABOUT MIN SIZE CONSTRAINT AND HOW TO ACTULLY GET THE CAMERA WHEN CAN SEE CODE).

Here are some examples of this working in practice. As you can see it is able to detect the face at various angles and with obstruction such as glasses. In these test images, notice the background is not neutral yet there are no incorrect faces detected.

(FINISH TALKING ABOUT FACIAL DETECTION AND HOW IT WORKS ETC)

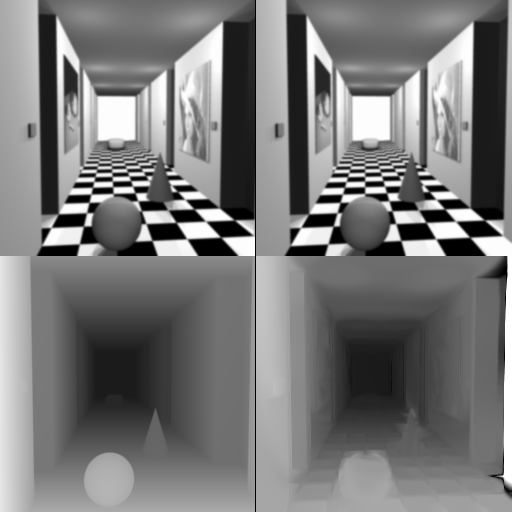
Now that the facial detection aspect of my project is working effectively. I have decided to devote some of my time to research how the number of people in a moving crowd can be counted and more specifically, how this can be done with the use of facial detection.

Potential method outline:

Pre processing

Facial detection

Frame

After the facial detection, there has to be some method in order to collect the relevant information about each face/person in the frame. A method I had previously researched I realised could possibly be used in my project. Disparity mapping refers to the apparent pixel difference or motion between a pair of still images. Much alike when you try closing one of your eyes and then rapidly close it while opening the other. Objects that are close to you will appear to jump a significant distance while objects further away will move very little. That motion is the disparity.

In this example, you can see the objects in the foreground are brighter, denoting greater motion and lesser distance. I believe this method could be used to determine distance away from the camera, speed of movement and distance moved. I plan to make a trail program in order to test this theory and measure its effectiveness.

Another method I have considered is to create a ‘counting line’ on the frame that is vertical/horizontal across the screen depending on expected direction of travel This height will need to be tested and decided while taking into account different heights of people. A point/array of points on a detected face are recognised and when these points/some of these cross/come in contact with the line, this is then counted as a person.