# 1 Requirement 1 : Correlation Detector

Simple correlation can be performed using a correlation of the template and image, all tests in this report used the obama.png as the template image. The results of using a simple correlation can be seen in figure 1. As can be seen this results is significant numbers of false positive detections, expecially in the flag section above the rows of people. Also many detections only cover half of a face.

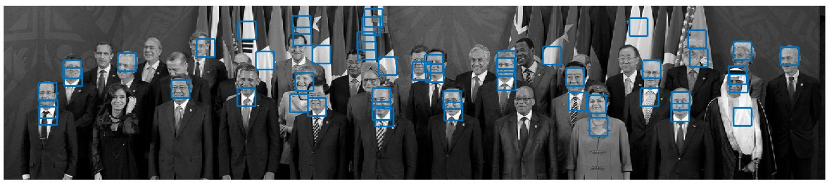

Figure 1: Results of using Simple Correlation on g20 image

To improve this a normalised correlation technique was implemented. The normalised correlation is defined in the coursework specification and is shown in figure 4 for reference. This can be simplified in a series of steps to make use of correlations in order to improve performance. Firstly the value of $x - m_x1$ can be precomputed as this is simple the template minus its mean. Next the numerator of the fraction can be simplified to $xy + m_x1m_y1 - xm_y1 - ym_x1$, each of these four terms can then be computed using a correlation. These four correlations are shown in figure 3 where tempalate and image are the vectorised image and template and templateMean is the mean of the template and imageMean is a matrix of the mean values at each tempalte location, computed using another correlation with a template of values 1/1024 where the template size is 32 x 32.

$$r = \frac{(x-m_x1)^T(y-m_y1)}{|x-m_x1||y-m_x1|}$$

Figure 2: Normalised Correlation

```
MxY = filter2(templateMean, image);
XMy = filter2(template, imageMean);
MxMy = filter2(templateMean, imageMean);
XY = filter2(template, image);
```

Figure 3: Numerator Compute Code

Now the numerator of the normalised correlation is computed we can turn to the denominator. The denominator of the normalised correlation equation is the standard deviation of the tempalte and part of the image under the template [**?** ]. The standard deviation of the template can be computed simply. For the part of the image under the template correlations can be used. The standard deviation can be rearranged to the equation shown in figure 4 where n is the number of pixels under the template, q is the squared sum of the pixels under the template and s is the sum of the pixels under the tempalte [1]. This can be computed using another three correlations to compute n, q and s as shown in figure **??** where the template is of size 32 x 32.

$$r = \frac{1}{n-1}(q - \frac{s^2}{n})$$

Figure 4: Standard Deviation

```
n = filter2(ones(size(32, 32), ones(32, 32), 'same');

imSquared = im .^2;
q = filter2(ones(32, 32), imSquared, 'same');

s = filter2(32, 32, image, 'same');

imStdDev = (q - ((s .^ 2) ./ n) ./ (n - 1));

imStdDev = sqrt(imStdDev);
```

Figure 5: Standard Deviation matlab

Using the normalised correlation the output of the face detection using obama.png as the template is shown in figure 6. As can be seen there is still a high number of false detections in the background of the image as well as in the tie area of some people. The next section will try and address these issues with an improved detector.

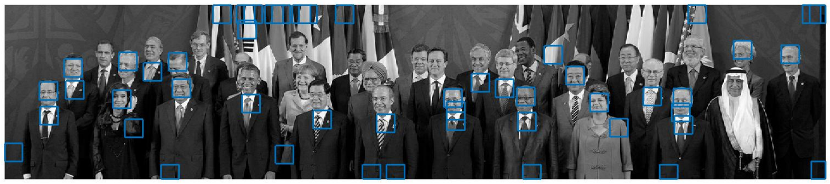

Figure 6: Results of using Normalised Correlation on g20 image

# 2 Requirement 2 : Improved Detector

The original normalised correlation detector achieves 27% overlap with the Viola-Jones face detection algorithm and this will be used as the performance measures for the improvements demonstrated in this section.

## 2.1 Guassian Blur

One improvement which was tried was to add a Gaussian blur using a correlation of a Gaussian template. The idea here was to remove some of the fine grain detain and high frequencies in the template image as these high frequencies may be causing some of the missed faces. This did increase the overlap to 29% the result can be seen in figure 7. The number of correctly detected faces is improved however there are still high numbers of false positives.

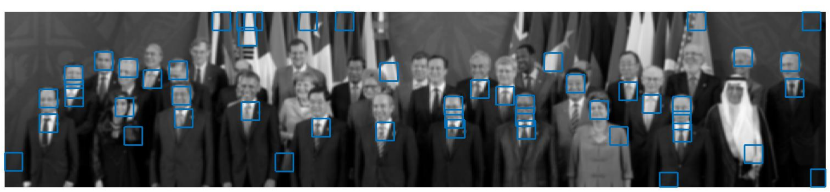

Figure 7: Results of using Normalised Correlation and Guassian Blur on g20 image

## 2.2 Colour Filtering

As many of the false detections are caused in the background of the image where the colours are not similar to skin tones a colour filter was implemented to try and reduce their occurrence. Using the training data supplied the upper and lower bounds for skin tone colours in red, green and blue where found. Any pixel which did not fall within these bounds was set to black. It was found that only applying the colour filter to the g20 image and not the template caused highest nubmers of detections. This achived an overlap of 50%, which is shown in figure 8. As can be seen in the figure there are significantly less false detections in the background of the image and also more correct detections as a result of reducing the search space down to the pixels which are within skin tone ranges.

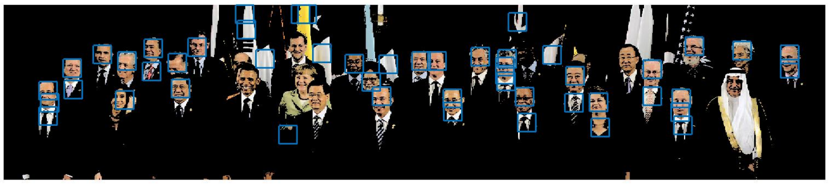

Figure 8: Results of using Normalised Correlation and Colour Filter on g20 image

## 2.3 Edge Detection

The final improvement on the original normalised correlation technique was using edge detection. The idea behind this was to simplfy the image down to just an edge detected image this means that the detection is just using the shape of the edges in the image as opposed to the colour or grayscale pixel value. It was expected that this would reduce the number of false detection is the background as the edges here are simple verticle lines which are not similar to those in the tempalate image. It was also expected that the number of correct detections would increase as the grey scale values are no longer being compared, so skin tone should not be an issue.

In order to create an edge detected image the Canny edge detection technique was applied. Firstly the Sobel Kernels are applied in two convolutions, for the horizontal and vertical. After this the gradients at each pixel are computed and are set to the closest of the two diagonals, horizontal and vertical. Finally the values are all thresholded to only leave the hard edges.

Using the edge detected image an overlap of 34% was achieved, this is shown in figure 9. As expected the number of false detections in the background of the image is reduced compared to the original normalised correlation and there are more positive detections.

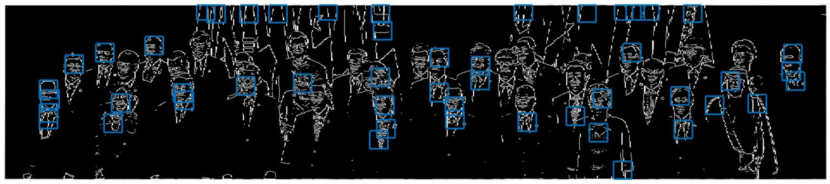

Figure 9: Results of using Normalised Correlation and Edge Detection on g20 image

### 2.4 Harr Features

The final detection technique which was tried was to move away from using a convolution of a template image and to use Harr features like the Viola-Jones algorithm does **??**. The Viola-Jones algorithm uses a cascade of simple classifiers using a Harr feature to detect faces. Therefore the use of a set Harr features to detect an face should yeild an improved overlap with the Viola-Jones algorithm.

In order to do this a seven Harr Features were used each of a fixed size, these were chosen by running all possible sizes of the two, three and four segment Harr features over a test set of faces and non faces. The features which gave the highest success rate when run on the test set were used in the final implementation. An integral image of the target image is created in order to compare the Harr features too. Each of the Harr features were used as the template for a convolution across the Wintegral image. The average of each pixel in each convolution is calculated giving an matrix where high values represent where more of the Harr features matched the image below. An example of the output when run on the g20 image is shown in figure **??**.

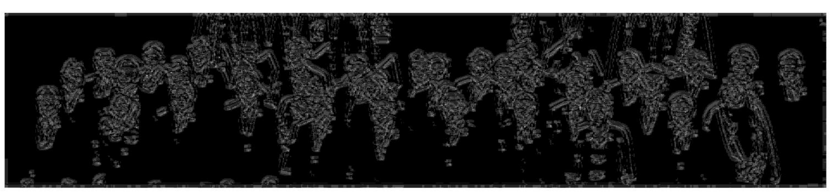

Figure 10: Results of using Harr feature technique on g20 image

## 3 Requirement 3

## 4 Requirement 4

Face Detection Normal - Normalised Corr - 27.21% Normalised Corr with Guass - 29.31% Normalised Corr with Colour - 50.16% //Only image colour filtered Normalised Corr with edge det - 34.38% //Only image edge detected My VJ - 49.16%

Face Classification

Nearest Neighborer - 5/18 SVM's = 4/18 Naive Bayes = 5/18

## References

[1] Matlab Tricks. Calculate standard deviation using minimal memory. `http://matlabtricks.com/post-19/calculating-standard-deviation-using-minimal-memory`, 2013.