Image Processing and Computer Vision Coursework 1

Tharidu Jayaratne – tj17908, James Lim – jl17056

Abstract

The aim of this coursework was to implement and experiment with conventional computer vision algorithms for object and shape detection. This project was broken down into 4 subtasks that describe our implementation of various algorithms on different images, the experimentations we conducted and our analysis.

Subtask 1: Viola-Jones face detection

The first visual object detection algorithm we used was *Rapid Object Detection using a Boosted Cascade of Simple Features* developed by Viola and Jones. This is a binary classifier built of several weak detectors, each being an extremely simple binary classifier. A cascade of weak detectors is trained to gain a desired precision using Adaboost, which combines multiple weak classifiers into a single strong classifier to detect objects.

We employed a strong classifier trained using AdaBoost for detecting frontal human faces using the provided XML file. We ran this algorithm on 8 test images containing faces and the results were as follows:

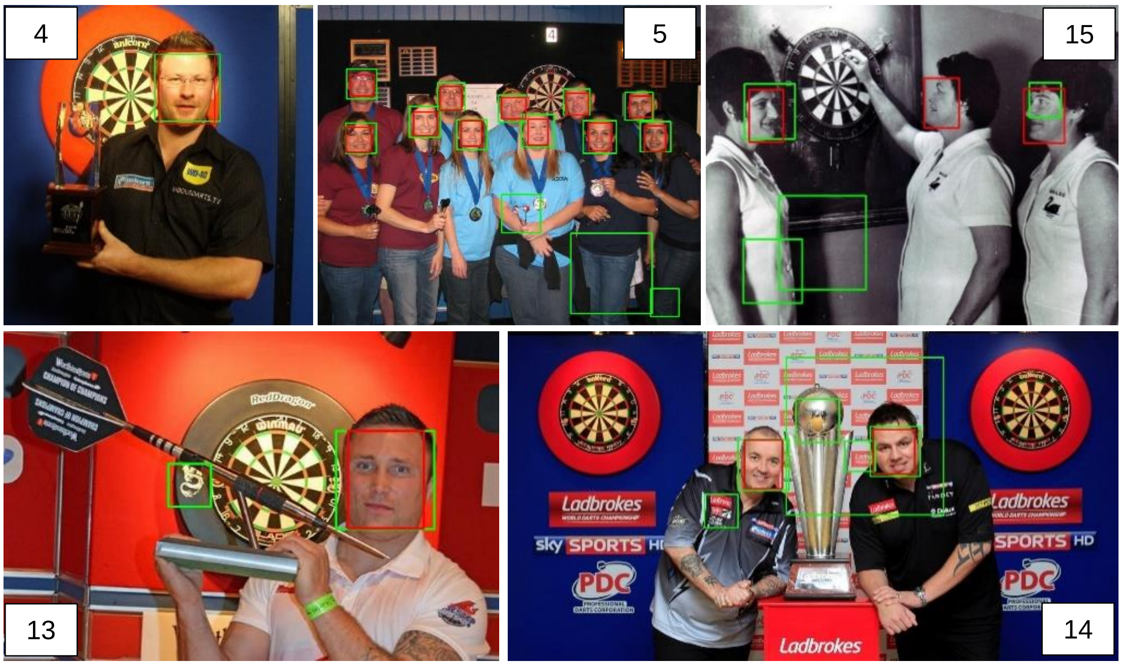

Figure 1: detected faces (green) and ground truth (red)

Table 1: Face detection results

Image	TPR	F1 Score	IOU
4	1	1	0.8
5	1	0.92	0.63
7	1	1	0.69
9	1	0.4	0.69
11	1	1	0.59
13	1	0.67	0.76
14	1	0.5	0.73
15	0.67	0.57	0.57
Average	**0.96**	**0.76**	**0.68**

The True Positive Rate (TPR) gives a measure of how sensitive the detector is. The F1-Score indicates the accuracy, considering the precision and sensitivity of the detector. The Intersection Over Union (IOU) indicates how much detected area corresponds to ground truth.

There are several cases in which the algorithm detects multiple overlapping instances, such as in *Image 14*. This can obstruct the assessment of a meaningful TPR, as it is difficult to know if the detected instance is of a face or a background object. To mitigate this, we used an IOU threshold of 0.5 to quantify what counts as a detected face.

Table 1 represents the results obtained from images where valid frontal faces were detected. Based on these results, it is almost always possible to achieve a TPR of 1 since the algorithm favours a higher hit rate at the cost of reduced precision, which increases the likelihood of detecting the true faces but lowers the F1-Score. Since the algorithm has been trained on frontal faces, it is not as capable of detecting faces shown at an angle. Therefore, it does not perform as well on *Image 15*. Finally, the high IOU ratios imply that the detector captures most of the features of a frontal face.

Subtask 2: Training a detector for Dartboards.

The second objective was to use the same algorithm as in *Subtask 1* to detect dartboards instead of faces. To do this, we constructed a detector using Haar-like features using OpenCV's boosting tool. First, we created a set of 500 positive samples from an example image of a dartboard and the utility **opencv_createsamples**, which randomly changes the contrast and viewing angle of the prototype image to create samples. Additionally, we provided the trainer with a set of negative samples to assess what is not a dartboard. Finally, we used the **opencv_traincascade** tool with the previous sets of images as parameters to train the detector via AdaBoost.

The training tool employs various stages to produce the classifier. At each stage, more weak detectors are added to the trainer, which contribute to the final decision made by the classifier. The performance at each stage can be measured by the TPR compared to the False Positive Rate (FPR). As shown in the figure below, the performance improves with each stage.

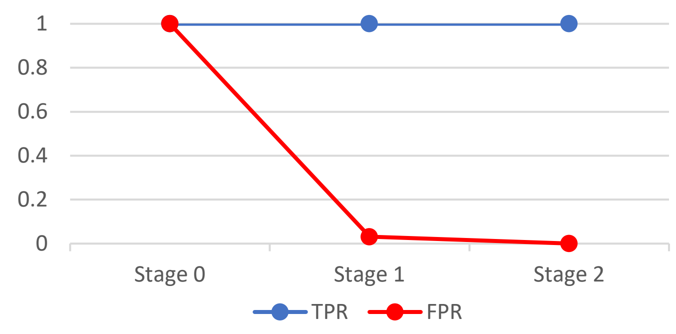

Figure 2: TPR and FPR at different stages of training

After stage 2, the FPR is very close to 0, while the TPR remains as 1. Adding more stages can further reduce the FPR but can also lower the TPR. In this case, this is the optimal number of stages for the classifier as it provides good generalisation with little overfitting.

To test the performance of this classifier, we applied it to 16 test images which contained dartboards. For each image, we calculated the TPR and F1-Score. Below are 3 examples containing the ground truth (in red) and the detections (in green).

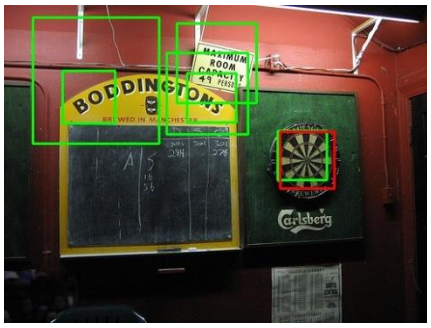

Figure 3: Trained detector on Image 3

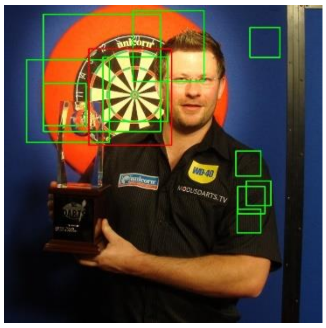

Figure 4: Trained detector on Image 4

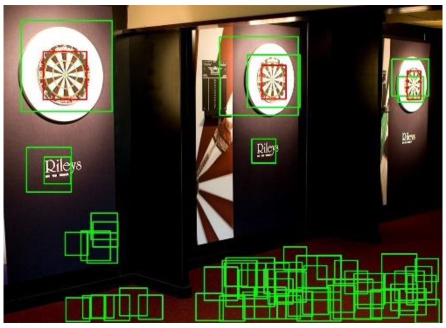

Figure 5: Trained detector on Image 10

As seen in the images, the classifier can successfully detect dartboards, even if they are partially occluded (*Figure 4)* or at an angle (*Figure 5)*. However, it detects many false positives where there are no dartboards, and often produces overlapping detections for a single dartboard (*Figure 4* and *Figure 5*). This method of classification leads to a high TPR at the cost of a low F1-Score due to the high number of false positives. *Table 2* shows the results across all images, which follow a similar trend.

There is a notable parallelism between the testing performance and the training performance of the detector. The training showed a TPR of 1 across all stages, which is reflected in the testing, achieving a TPR of 1 on all images but one.

Although the detector can find dartboards in an image, the performance and accuracy can be improved greatly by reducing the number of false positives. This can be done by filtering the detected dartboards according to their shape configurations using a Hough transform.

Table 2: Detector performance

Image	TPR	F1 Score
0	1	0.17
1	1	0.22
2	1	0.25
3	1	0.29
4	1	0.18
5	1	0.12
6	1	0.2
7	1	0.11
8	1	0.18
9	1	0.2
10	1	0.1
11	0.33	0.33
12	1	0.4
13	1	0.2
14	1	0.09
15	1	0.4
Average	**0.96**	**0.22**

Subtask 3: Integration with Shape Detectors

We combined the output from the Viola Jones detector in Subtask 2 with the Hough transform algorithm. This was to ensure we consider shape configurations of the dartboard.
The first task was to implement an edge detecting algorithm to remove any unnecessary edges and find object boundaries. We did this by calculating the image gradient (encoded by its magnitude and direction) using the partial derivatives of an image in the x and y directions. This was obtained through convolution with Sobel operators. The output results in a gradient that points in the direction of most rapid increase in intensity as x and y changes. Using these partial derivatives, we calculated the image magnitude and direction.
We then had to threshold the image magnitude to extract the strongest edges. We used 30 as we found it to produce the best outlines of a dartboard. Once we had our threshold image, we used the direction to calculate the coordinate where each pixel was pointing towards and intersecting. We made an accumulator to track the number of intersections for each point in the parameter space using the coordinates calculated. The number of points intersecting the coordinate would increase by one for every intersection. Finally, we mapped each value in the accumulator, onto an output matrix, normalizing it and then extracting points with an assigned radius that has the highest number of votes. The value we used to extract the Hough centers was 255, as we did not want to display multiple circles by lowering the threshold. Using these Hough centers, we translated the Viola-Jones detections that had the best IOU with our Hough squares. We combined our Hough algorithm with Viola-Jones in this manner as we believe that the Hough centers had better precision and it can give robust detection under noise and partial occlusion, based on the increase in F1 scores in *Table 3* after combing Hough with Viola-Jones. Furthermore, Viola-Jones considers features other than shape configuration. After applying this algorithm to the test images and examining the results from *Table 3*, we can draw a few conclusions:

Figure 6: Threshold image, Hough Space and detection for Image 8

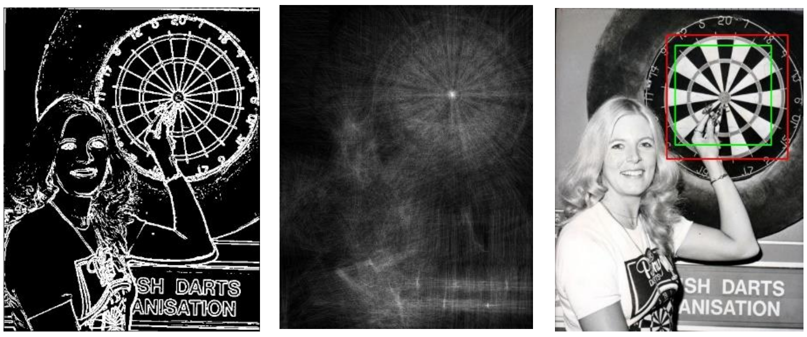
Figure 7: Threshold image, Hough Space and detection for Image 9

Table 3: Combined detector performance

Image	TPR	F1 Score	Increase in F1 Score
0	1	1	0.83
1	1	1	0.78
2	1	1	0.75
3	0	0	-0.29
4	0	0	-0.18
5	1	1	0.88
6	1	1	0.8
7	0	0	-0.11
8	0	0	-0.18
9	1	0.67	0.47
10	0.33	0.4	0.3
11	0	0	-0.33
12	0	0	-0.4
13	1	1	0.8
14	0.5	0.5	0.41
15	1	1	0.6
Average	**0.55**	**0.54**	**0.32**

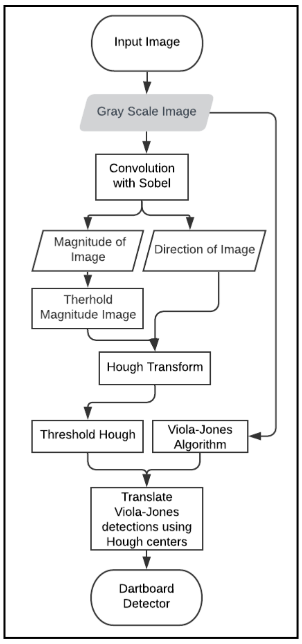

Figure 8: Flowchart describing the combining process

Merits, as shown in *Figure 7*:

- Reduced False Positives.
- Higher Precision.
- Increased True Positives.

Limitations, as shown in *Figure 6*:

- Poor performance in dimly lit and blurry images.
- Difficult to extract all dartboards without having concentric circles and constant change of thresholds.
- If either algorithm doesn't detect any dartboard, no detection will be made.

Subtask 4: Improving the detector

The main complication with the detector thus far is detecting multiple dartboards in a single image which do not create bright enough pixels in the Hough space. A way to detect these dartboards would be lowering the threshold at which an instance is detected in the Hough space. However, doing this creates many instances close to each other, since there are many pixels in the same place which would exceed the threshold.

- A high threshold prevents duplicates but cannot find all dartboards.
- A low threshold can find more dartboards but creates duplicates in the same space.
- When lowering the threshold, duplicates appear before other dartboards are found.

A solution to this problem is dividing the Hough space into a grid and finding a maximum of one circle per section. Once an instance is found in one section, the algorithm continues searching from the start of the next section. We decided to divide images into 6 sections (3 divisions on the X axis and 2 on the Y axis), as we believe this will produce sections which will not contain more than one dartboard in most cases.

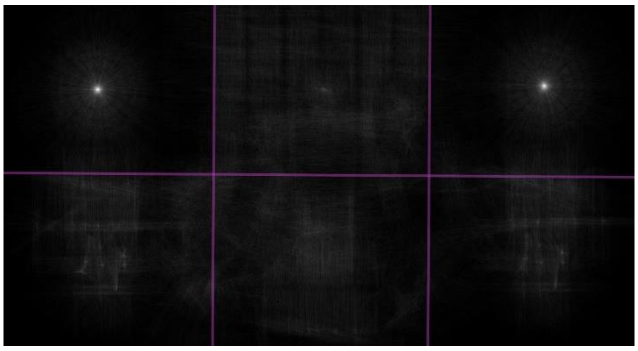
Figure 9: Divided Hough Space

This approach allowed us to lower the Hough space threshold from 255 to 200 without resulting in duplicate detections for a single dartboard. Additionally, the lower threshold enabled detections for dartboards which did not produce bright enough pixels in the Hough space.

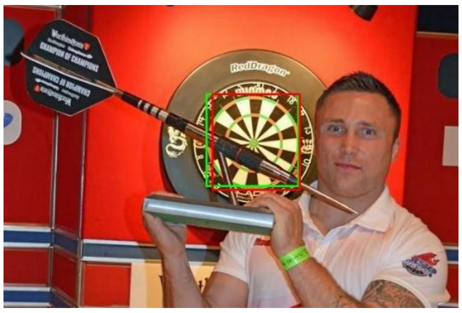

Figure 10: Final detector for Image 13

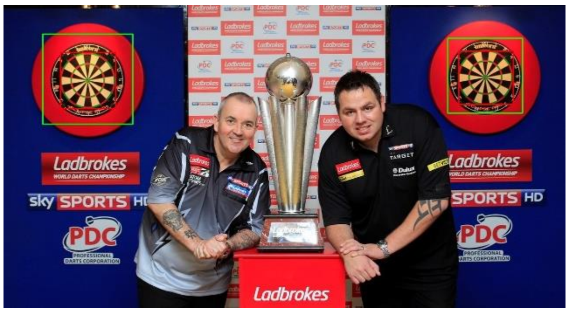

Figure 11: Final detector for Image 14

A drawback from this approach is the possibility of a Hough center being located at the boundaries between section. This could lead to more than one instance being detected for a single dartboard. Additionally, if there is more than one dartboard per section, it will only detect one of them.

Table 4: Final detector performance

Image	TPR	F1 Score	Increase in F1 Score
0	1	1	0
1	1	1	0
2	1	1	0
3	1	1	1
4	1	1	1
5	1	1	0
6	1	1	0
7	1	1	1
8	0.5	0.67	0.67
9	1	1	0.33
10	0.67	0.8	0.4
11	0	0	0
12	1	0.5	0.5
13	1	1	0
14	1	1	0.5
15	1	1	0
Average	**0.89**	**0.87**	**0.34**

As seen in *Table 4*, the F1-Score shows a significant increase compared to the previous version since the lower threshold allows for more flexibility when extracting Hough centers. This is especially noticeable in cases where there are multiple dartboards and images which are dimly lit. Our final conclusions were as follows:

Merits:

- Decreased False Negatives and increased True Positives.
- Can detect multiple dartboards in a single image.
- Can detect dartboards in dimly lit images.

Limitations:

- If either algorithm doesn't detect any dartboard, no detection will be made.

Contribution out of 2: Tharidu Jayaratne = 1, James Lim = 1

[illegible] [illegible]