Dartboard Challenge: Image Processing and Computer Vision

Sub-task 1: The Viola-Jones Object Detector

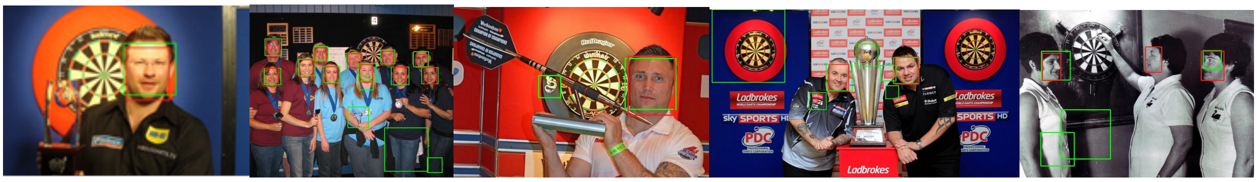


Image No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Recall/TPR	0.00	1.00	1.00	1.00	1.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	1.00	1.00	0.66	1.00
F1 score	0.00	1.00	1.00	1.00	1.00	0.44	0.00	1.00	0.00	0.40	0.00	1.00	1.00	0.67	0.57	0.84

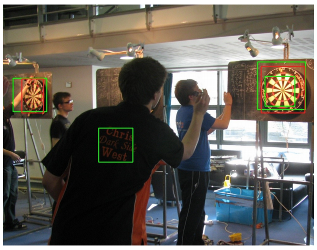
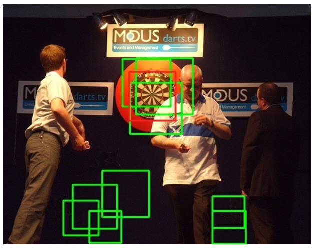
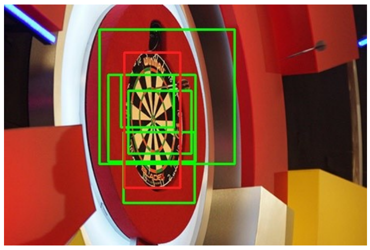
To begin with, we needed to find a suitable threshold for the detected Viola-Jones bounding boxes. When looking through the values being outputted, it became clear that this could be a rather arbitrary number. An example being Dart15.jpg [shown above]. While the Viola Jones algorithm was only meant to detect frontal faces, it was able to pick up the two faces on the far left and right. Despite detecting them, they were not complete and required fine tuning the threshold to allow them as a True Positive. We started at a 0.5 threshold, but it missed a few faces. We lowered it to 0.4 and got a near 100% TPR across all images, even in image 15 where all the faces are at an angle, 2 of 3 were detected.

A result of this shows that with a lenient enough detector and threshold, you can achieve a 100% TPR. The problem with this would be detecting many False Positives, lowering the Precision. To see this we look at the F1 score, a measure of precision and TPR [Hence Recall] combined. Across the table, the F1 score is varies heavily. This is due to the leniency of the threshold applied to the Haar features in the images. Too many False Positives spoils the broth. A good example of this is image “dart14.jpg” [seem above]. Both Boards were successfully identified but many more were found towards the center. Despite achieving a Recall of 1, the F1 score was 0.4.

The way to circumnavigate this would be through appropriate training of the Haar features. The detector currently detects many False positives. With a better classifier this error rate should start to be reduced. In order to obtain a better classifier, a couple of methods can be employed. Firstly, the amount of training data for the adaboost training could be increased. More training data allows for more flexibility and robustness of the classifier as it will have more to compare. This can be applied to both the positive sample and the negative samples, each allowing different methods of comparing a weak classifier. Secondly, Incorporating a different boosting and classifier algorithms such as Gradient Boosting which can adapt to different distributions. By experimenting with the training data we can draw a conclusion on which will be better.

In conclusion, the Viola-Jones is a good system for detecting a given classifier. It was able to consistently find the frontal faces as well as partially angled faces with varying success. It’s downfall is that it is not an overall robust system. This lead to many false positives in certain images. Without a solid system to detect where to look for the features, the Viola Jones algorithm can become easily confused.

Sub-task 2: Building & Testing your own Detector

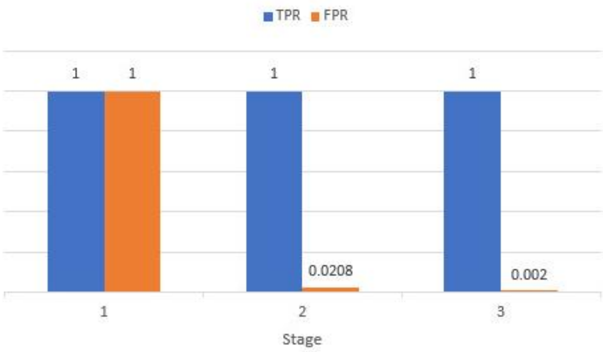


Illustration 1: Adaboost training performance

The figure above is the chart depicting FPR and TPR during the training stages. Once the training has gone through one stage, the FPR drastically decreases. This is due to stage 0 not having a classifier yet thus the Viola-Jones algorithm accepts all images. Once the third stage has finished, the FPR is almost non-existent. Meanwhile the TPR does not decrease across all stages. This shows the adaboost training has created a progressively better classifiers by selecting and adding features. Here would be a good stage to end on as going any further increases the risk of over-fitting without any major benefit from the given data set.

Given more training data, it could possibly identify further features to be added to the classifier. More variety in the Positive data set, such as angling the board, could create a more robust classifier.

Performance Testing

Overall, almost every board was successfully identified but the precision left much to be desired. In many cases throughout the test data a dart board would be detected multiple times. Compared to the performance seen from the training set, it is very lacking. There were many False Positives throughout every image, as indicated by the precision in Table 1. To reduce the False Positive Rate, the learning would need to be improved. With the training data given, the FPR was already too low to proceed in refining the classifier.

However, there are a couple of other ways to improve this method. Firstly, we could alter the program to take these multiple detected Positives to create a new, single positive detection by averaging out the bounding boxes which overlap. Before we could have multiple detected boxes for a give board, each only showing a small area of the board. By combining these, we could create a single detected area covering most if not all the Board.

Alternatively, should a detected box overlap with another significantly, we could have the smaller of the two be removed as that area is already detected. Using the IOU function already created, this would be a simple matter and provide a more consistent detection. Combining these two techniques could be a through way to implement the Viola-Jone algorithm in noisy images.

Image No.	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**Avg**
Recall/TPR	1.00	1.00	1.00	1.00	1.00	1.00	0.00	1.00	1.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00
Precision	0.2	0.25	0.13	0.2	0.14	0.16	0.11	0.07	0.50	0.20	0.16	0.33	0.20	0.20	0.12	0.50	0.22
F1 score	0.33	0.4	0.22	0.33	0.5	0.29	0.00	0.16	0.66	0.33	0.27	0.50	0.22	0.33	0.21	0.67	0.35

Table 1: Viola-Jones example data

Sub-task 3: Integration with Shape Detectors

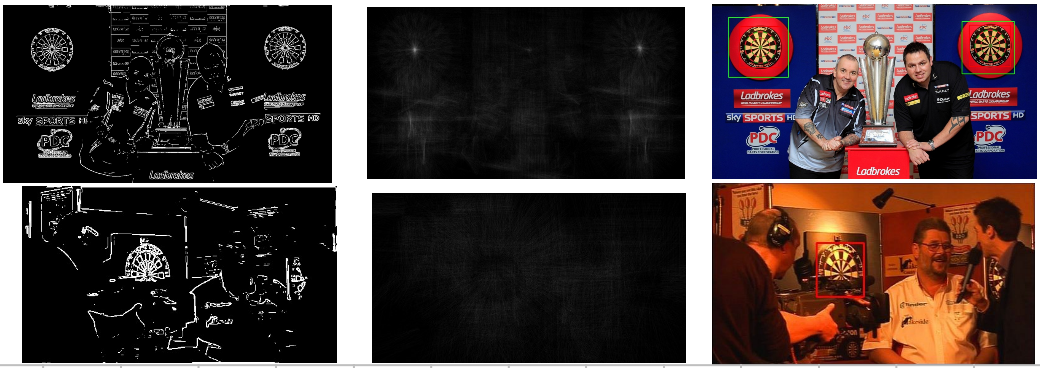


Image No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Avg
Recall/TPR	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50	1.00	0.33	0.00	0.00	1.00	1.00	1.00	0.74
TPR Diff	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.50	0.00	0.33	-1.00	-1.00	0.00	0.00	0.00	-0.01
F1 score	1.00	1.00	1.00	1.00	0.66	1.00	1.00	1.00	0.66	1.00	0.33	0.00	0.00	1.00	1.00	1.00	0.73
F1 Diff	0.66	0.60	0.78	0.67	0.11	0.71	1.00	0.84	0.00	0.67	0.06	-0.50	-0.22	0.67	0.79	0.33	0.41

Pros and cons

+ High confidence in detected boards due to the combination of detection methods.

+ With threshold for IOU between hough circles and Viola-Jones detection allows for precise detection where only viola-Jones would detect multiple boxes.

+ Multiple boards detected with ease

- Only detects front faceing boards

- If one method does not detect the board, it will cancel out the other’s detection, should there be one

- Small boards or Heavily occluded boards are tough to be identified

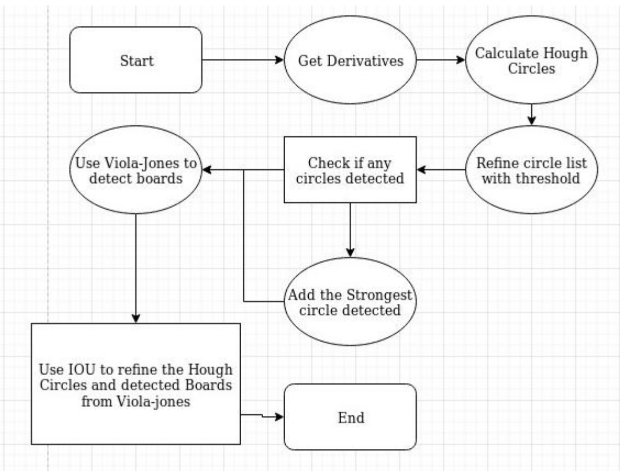


Illustration 2: Board Detection Flow diagram

Methodology

+ With the flow of data being so clean, we were able to print of values for testing purposes regularly

+ This also allowed for easy debugging as the method grew as the complexity did not increase massively, only how we handled each output

+ By printing values regularly, we could curate the values to the data set and fine tune thresholds easily

+ Refinement method creates a strong confidence rating in a detected board where one single method would have low precision

+ This plug and play method allowed for testing and altering of individual elements as time went one without needing large overhauls to the code

- Unable to use one method to detect a board, allowing one method to cancel the other

- does not prevent multiple detection of the same board, needed to refine threshold as best as possible to mitigate this

- Hard to beat one single method's TPR, instead it increases the confidence/precision of both.

Sup-task 4: Improving your Detector

Introduction

Our approach to improvements was simple: How do we take what we have made and adjust it for better detection? Our initial detector combined Hough Circles and Viola-Jones via refining the detected areas. From there we looked to see how to improve what we had.

Best of the Rest

While testing different thresholds for the Hough circle identification, particularly with occlusion, we found that often we would have to set the too low to be useful for other images. To counter this, we set the threshold to a value appropriate for the majority of images, then had a variable store which was had the most votes to be used as a back-up in case no circles beat the threshold. This allows for consistent detection which, when combined with the Viola-Jones detection, works nicely of lightly occluded boards.

Noise Reduction

Images and picture can have a lot of noise for many reasons. This is one of the biggest issues when it comes to line detection. Enough ‘noise, can allow for false positive edges, which results in false positive Line and circle Detections. In order to prevent this while maintaining the desired edges, we used a light Gaussian blur. This removes most areas with small differences in them while keeping the harder edges we are looking for.

Canny Vs Sobel

Another minor tweak we were able to perform was substituting the edge detection methods we used. Initially we employed the methods created during the coin challenge: Sobel. While this worked well, particularly when combined with a threshold function to set an edge as 0 or 255 in gray scale, OpenCV has other options which build upon Sobel, namely Canny. Canny uses sobel for angle detection but has a slightly more sophisticated method for detecting whether there is an edge. Also, it allow for more options such as threshold and better gradient functions for more processor time. This gave a cleaner edge detection making the Lines and Circles detection more consistent.

Hough Lines

Once done with the previous tweaks, we looked for other methods to detect the dart boards. To do this, we tried the Hough Lines Algorithm. The Rational behind this was: a key feature of a dart board would be the lines that intersect at the center. With the detected Lines, we added them to a black image and allowed the brightness to overlap, similar to the Hough Space images. This gave us the positions of where the lines intersected. Once thresholded we were able to detected the center of dart boards, in theory.

Upon testing however, we found thresholding the intersections and lines to be too reliant on the image itself, and finding a single best fit proved to be an ineffective method when compared to the Hough circles. Instead, we went with the Hough Lines as a last resort should a circle not be detected.

Images used are the same as Task 3, The values are slightly altered to have a higher F1 score but ultimately it was not a significant margin.