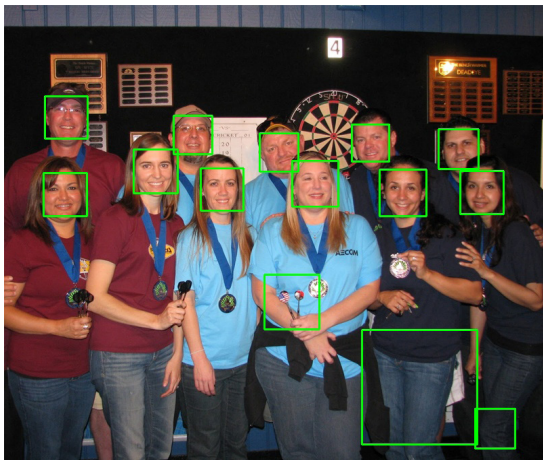
Image Processing and Computer Vision

Subtask 1 : The Viola-Jones Object Detector

The objective of the first subtask is to test the performance of the face detection system and the provided classifier. The resulting images are shown below:

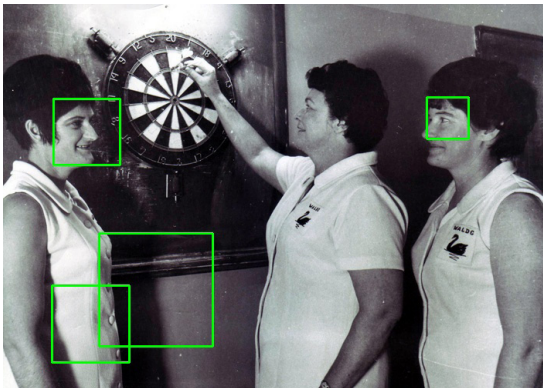
dart4.jpg



dart5.jpg

dart13.jpg

dart14.jpg



dart15.jpg

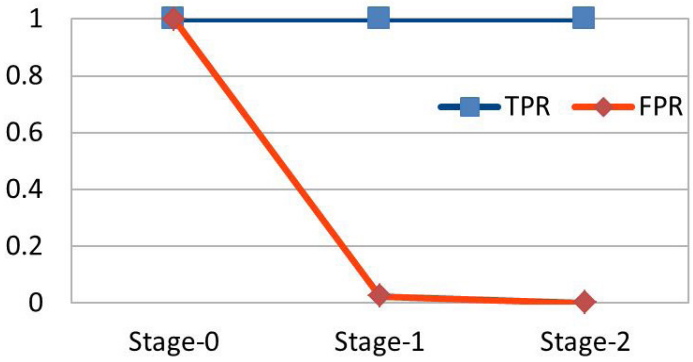
A possible way of measuring the performance of a binary classifier such as our face detection system is by calculating the true positive rate(TPR). This is defined mathematically as TPR=TP/(TP + FN) where TP and FN stand for "true positive" and "false negative" respectively. In simple terms, it is the number of detected faces(true positives) divided by the total number of faces(true positives and false negatives).

This metric, which is also known as recall or sensitivity of the detector, can sometimes be difficult to measure accurately due to the fact that a large sample is required for this to be a true estimate of the probability that an arbitrary member of a class is detected correctly. For example, in the first picture(-dart4.jpg), since there is only one face which was correctly detected(i.e. no false negatives/undetected faces), we have a TPR=1/1=100%. For pictures dart5.jpg and dart15.jpg, TPR is equal to 11/11=100% and 2/3=66.67% respectively. In addition, depending on the picture, the object being detected and the intended use of the detector, ground truth can sometimes be difficult to determine, hence TPR calculations vary accordingly.

The true positive rate is not the only metric by which the performance of a detection system can be estimated. Since TPR does not take into account the number of objects that were falsely classified as faces(i.e. false positives or FP), we can always increase TPR by increasing the sensitivity of a system. In fact, a system that classifies every possible object in an image as a face, would still have a TPR of 100% but of course that would not be a very useful classifier because it would have very low precision. This suggests that another metric is needed which would also take into account the false positives. That is the F1-score and it is defined as F1=2TP/(2TP+FN+FP). For example, for dart15.jpg this is F1=2*2/(2*2+1+2) = 4/7 = 57.14%. To calculate this, we have written a function that attempts to match each rectangle returned by our system with a defined set of rectangles indicating where each face truly is in a picture(ground truth). A pair of rectangles are matched when the ratio of the area of their intersection to the total(union) of their areas is greater than 0.3 i.e. I/(A1+A2-I) where A1 and A2 are the areas of the first and second rectangles respectively and I is the area of their intersection. The number of matched pairs of rectangles is our true positive value. The number of unmatched rectangles in our ground truth array is our false negative value and the number of unmatched rectangles, from those that were returned by our system, is the false positive value.

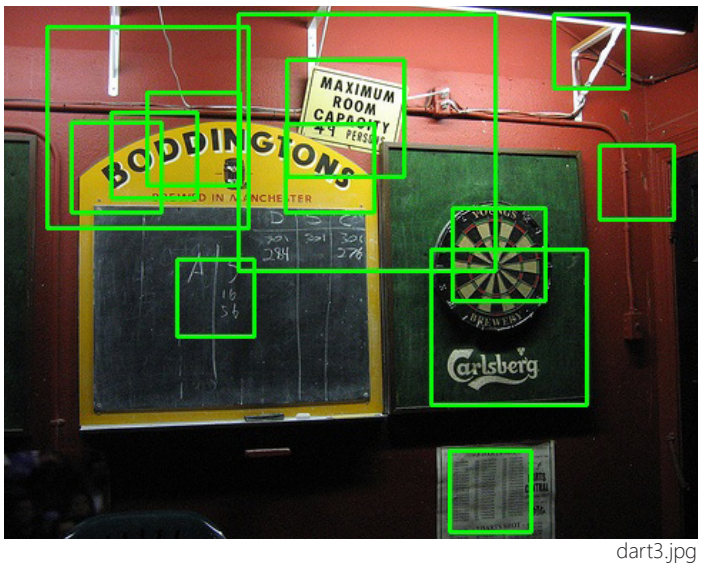
Subtask 2 : Building & Testing our own Detector

The objective of the second subtask is to use two OpenCV tools, one that will produce a set of 1000 positive training sample images of dartboards from a given image and another for training the classifier in order to detect dartboards in images, instead of faces using the generated positive images and the given negative images. During the different training stages, the boosting provides statistics about the performance of the classifier in each stage, TPR and FPR. Here are the results we obtained when running the training procedure:

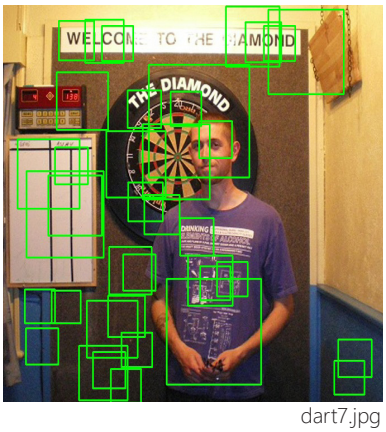


Observing the values obtained we can see that, initially, the detector is highly sensitive, meaning a high TPR as well as a high FPR. As the training procedure proceeds to the next training stage, it increases the number of features to be used by the classifier, which in turn, will discard false positives without altering the amount of true positives. This is similar to a constrained optimization problem, where a large set of objects is initially detected and we optimize this set by getting rid of false positives. The next step of this subtask required that we run the dartboard detector using the cascade, measure its current performance and visualize the results. Below is a list of the F1 scores for all the example images plus four example outputs with bounding boxes around detected objects:

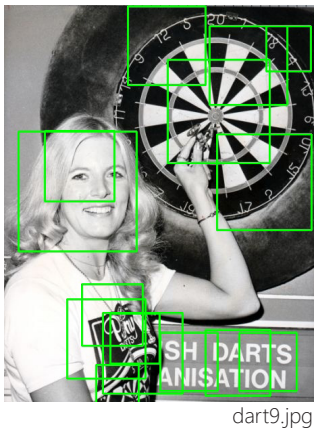
Image	F1	TP	FP	FN
dart0	0.11	1	16	0
dart1	0.15	1	11	0
dart2	0.14	1	12	0
dart3	0.14	1	12	0
dart4	0.12	1	12	0
dart5	0.13	1	15	0
dart6	0.07	1	14	0
dart7	0.05	1	25	0
dart8	0.13	1	37	0
dart9	0.12	2	28	0
dart10	0.09	2	13	0
dart11	0.20	3	63	0
dart12	0.20	1	7	1
dart13	0.11	1	8	0
dart14	0.08	1	17	0
dart15	0.22	1	7	0
Overall F1 Score	0.129			

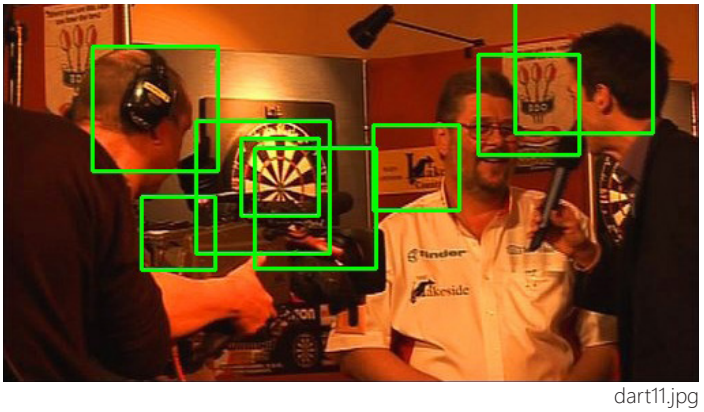


dart3.jpg



dart7.jpg


dart9.jpg


dart11.jpg

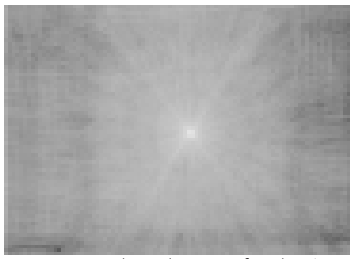
As we can observe from the output images, all of the dartboards were correctly detected(TP) but there exist a lot of FP detections as well. So given the graph above, we can now say for certain that all of the output have a TPR=1 as correctly predicted. We experimented by running the training procedure for another fourth stage and we got better results with decreased number of FP detected, but in some cases we could observe an increased number of FNs. Comparing the performance of our detector with the graph produced above, it is clear that it does not reflect its true performance on a different data set. Given the values of the TPR(1) and FPR(0.00067) obtained from the graph for the final stage of the classifier, the results were different from what was expected since high TPR and very low FPR should produce an almost perfect detector. Therefore, the plot is in fact useless in the sense that it is biased based on a specific (training) data set.

Subtask 3 : Integration with Shape Detectors

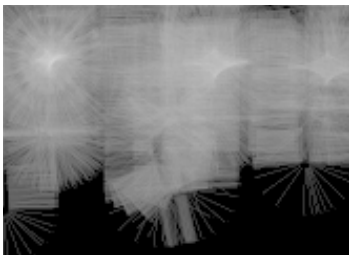
Below we can see the resulting images for this subtask, including their edges, 2D Hough space and the final detections:

Edges of dart1.jpg

Edges of dart10.jpg



2D hough space for dart1.jpg



2D hough space for dart10.jpg

Final detection dart1.jpg

Final detection dart10.jpg

Next is the table showing the F1 score for each image and the overall F1 score. We can see a significant increase in performance due to the implementation of the Concentric Circles Hough transform:

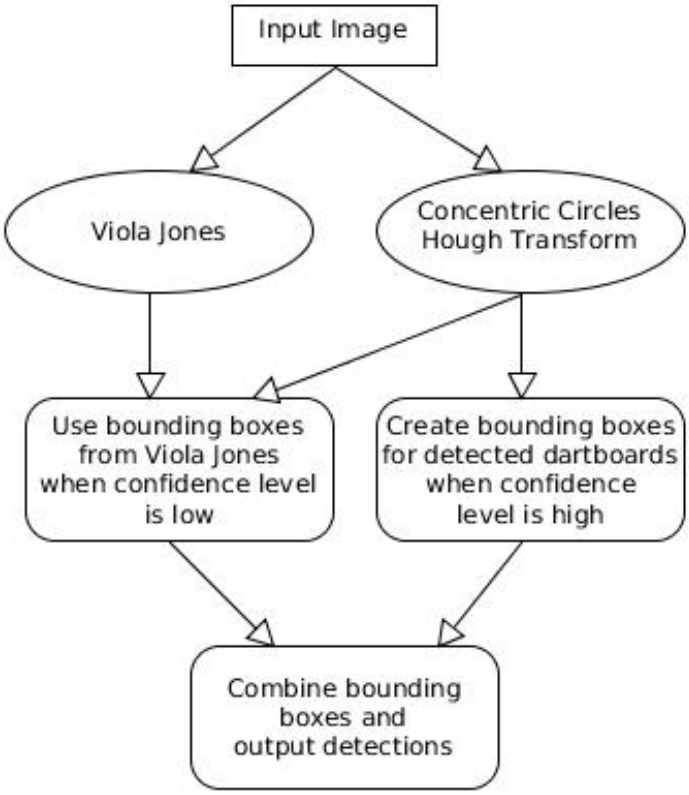
Image	F1	TP	FP	FN
dart0	1	1	0	0
dart1	1	1	0	0
dart2	1	1	0	0
dart3	0	0	0	1
dart4	1	1	0	0
dart5	1	1	0	0
dart6	1	1	0	0
dart7	1	1	0	0
dart8	0.67	1	0	1
dart9	0.50	1	1	1
dart10	0.80	2	0	1
dart11	0.67	1	0	1
dart12	0	0	0	1
dart13	1	1	0	0
dart14	0.67	2	2	0
dart15	1	1	0	0
Overall F1 Score	0.769			

Pros:

- Reasonably high F1 score
- Partially occluded dartboards are detected
- Near-perfect bounding boxes for front-facing dartboards

Cons:

- Dartboards are not detected when viewed from an angle
- Lighting affects edge detection thus preventing hough transform from detecting concentric circles (e.g. dart3.jpg, dartboard frame specular highlights)



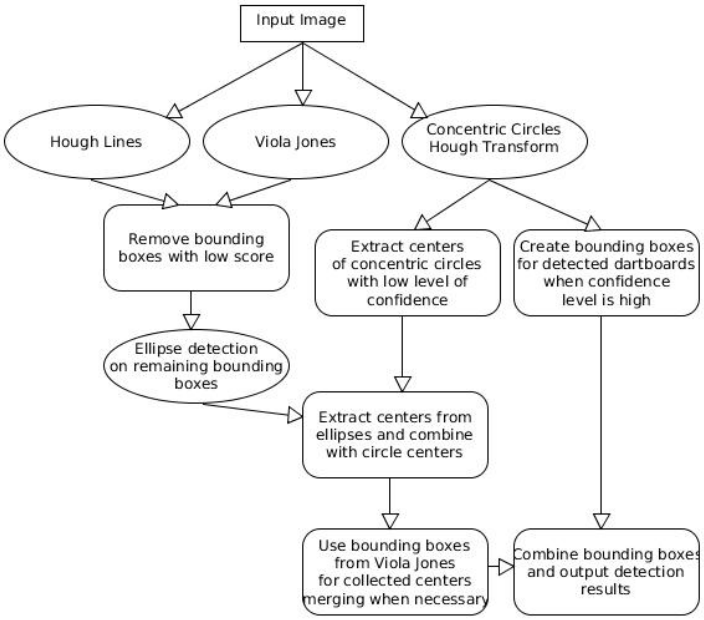
Flow diagram representing the order of operation of our detector

After testing various methods to combine the Viola-Jones detector with our concentric circle Hough Tranform, we decided on the following implementation:

- We use concentric circle Hough transform to detect the centers of the dartboards. Since the radii returned are accurate in most cases, we decide whether to use one of the Viola-Jones' bounding boxes or create a new one based on the data obtained. This is done using a threshold on the total accumulator value outputted by the Hough transform(here referred to as confidence level).
- For the centers, of which the radius was not accurate(failed the confidence level test), all Viola-Jones bounding boxes close to each center are combined to create new bounding boxes using a weighted average(based on the distance from the detected center) of the top left and bottom right corners of each box.
- Canny edge detector was used as input to the Hough transform since the thinner edges result in a much more accurate detection.

Subtask 4 : Improving your Detector

- Due to the fact that some of the dartboards from the data set provided were viewed from an angle and when projected on a 2D plane (image) resemble the shape of an ellipse, we decided on implementing an Ellipse detector([1] and [3]), based on the fact that the concentric circle Hough Transform had trouble detecting ellipses. Ellipse detection is a computationally difficult and timely task, therefore OpenMP was used to take advantage of the full processing power available, thus increasing performance.
- The algorithm used was required to process a large number of possible major axes. In order to limit this large computation, randomly selected major axes were chosen as candidates to be processed [2].
- Eventually, to further speed up the overall detection, we made use of Line Hough Transform to exclude bounding boxes from the collection outputted by the Viola Jones detection, by assigning a score to each bounding box and discarding the boxes with a low score.



Flow diagram of our improved implementation

Final detection dart8.jpg

Final detection dart12.jpg

Below is a table containing the F1 score each image plus the overall F1 score for all the images, based on our now improved implementation. There is a significant increase in performance and efficiency based on the fact that it can now detect ellipses.

Image	F1	TP	FP	FN
dart0	1	1	0	0
dart1	1	1	0	0
dart2	1	1	0	0
dart3	0.67	1	1	0
dart4	1	1	0	0
dart5	1	1	0	0
dart6	1	1	0	0
dart7	0.67	1	1	0
dart8	1	2	0	0
dart9	0.50	1	1	1
dart10	1	3	0	0
dart11	0.67	1	0	1
dart12	1	1	0	1
dart13	1	1	0	0
dart14	0.67	2	2	0
dart15	1	1	0	0
Overall F1 Score	0.886			

Final Pros(additionally to the previous pros):

- Higher F1 score; Improvement to the previous built.
- Can detect dartboards from different angles; TPR is always 1.
- Eliminated most of the FNs existing in the previous implementation.

Final Cons:

- Could not eliminate all FP and in some cases introduced more.
- Random aspect of the detector does not ensure consistent results.

Possible further improvements

- Harris corner detection could be used to further eliminate the FPs.
- Threshold image based on the values of the different colour channels, to further distinguish the dartboard and improve on edge detection.

References:

[1]"A New Efficient Ellipse Detection Method" (Yonghong Xie Qiang , Qiang Ji / 2002)

[2]Random subsampling inspired by "Randomized Hough Transform for Ellipse Detection with Result Clustering" (CA Basca, M Talos, R Brad / 2005)

[3]"Ellipse Detection using 1D Hough Transform" - http://uk.mathworks.com/matlabcentral/fileexchange/33970-ellipse-detection-using-1d-hough-transform