Dartboard Detection

Rutterford, Ainsley
ar16478@my.bristol.ac.uk

Waugh, Harry
hw16471@my.bristol.ac.uk

First Subtask

After examining the images and running them through the Viola-Jones classifier, we proceeded to compare them to the annotated ground truth images, i.e the actual locations of the faces. By comparing these two image sets, we calculated the 'true positive rate' or 'recall' value for each image. The F_1 score can be calculated using the following equations:

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \qquad precision = \frac{tp}{tp + fp} \qquad recall = \frac{tp}{tp + fn} \tag{1}$$

Where tp, fp, and fn are the amounts of true positives, false positives, and false negatives respectively. The calculated recall values for the `dart5.jpg` and `dart15.jpg` images were 100% and 100% respectively. A true positive rate of 100% gives the impression that the Viola-Jones classifier has performed very well on these images. However, this value does not include the number of false positive values (i.e. the number of objects that were incorrectly identified as faces). It is also important to note that in any detection task it is trivial to achieve a 100% true positive rate, for example by identifying every possible size and position of a group of pixels in the image as a face. A detector like this would obviously be unusable as it would also have an extremely high false positive rate. Another way to achieve a 100% true positive rate is by creating an algorithm that is incredibly specific to one image. The issue with this approach is that it would result in overfitting the model, allowing the classifier to perform perfectly with the image it was trained for, but possibly very poorly for all other images.

It may sometimes be difficult to assess the true positive rate accurately as it may be unclear as to whether or not a face should be detected. For example in `dart15.jpg`, should the faces shown from the side be detected? And if so how far can they turn until they should not be detected anymore? We decided that since the classifier provided was trained for frontal faces, we would only allow a face showing both eyes to be considered as a face. This meant that the `dart15.jpg` image did not contain any faces resulting in a true positive rate of 100%.

An F_1 score can be used to compare the success of a model against the 'ground truth'. We stored the information for each ground truth rectangle containing a face in a `.csv` file. We then wrote a piece of code that would calculate the F_1 score automatically given a `.csv` file and a list of the detected rectangles. In order to decide whether or not a detected rectangle was in fact a true positive, we used the percentage overlap of the detected rectangle and the ground truth rectangle. This meant that we could use a percentage tolerance as opposed to a fixed pixel tolerance, allowing us to use the same tolerance value with images of different resolutions and images with smaller or larger faces (or dartboards later on). If multiple rectangles were similar to the ground truth rectangle, the rectangle with the highest percentage overlap would be chosen and the rest would be treated as false positives. After running this code on the images provided, the average F_1 score for every image was 0.18.

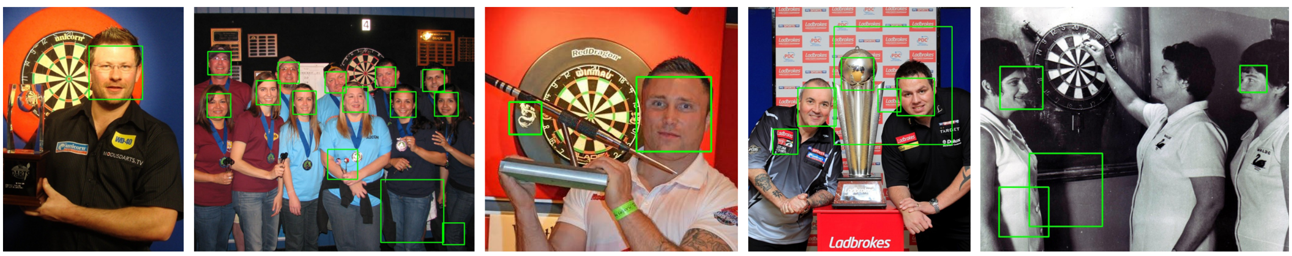

Figure 1: Faces detected by the Viola-Jones classifier. The images `dart4.jpg`, `dart5.jpg`, `dart13.jpg`, `dart14.jpg` and `dart15.jpg` are shown from left to right.

Second Subtask

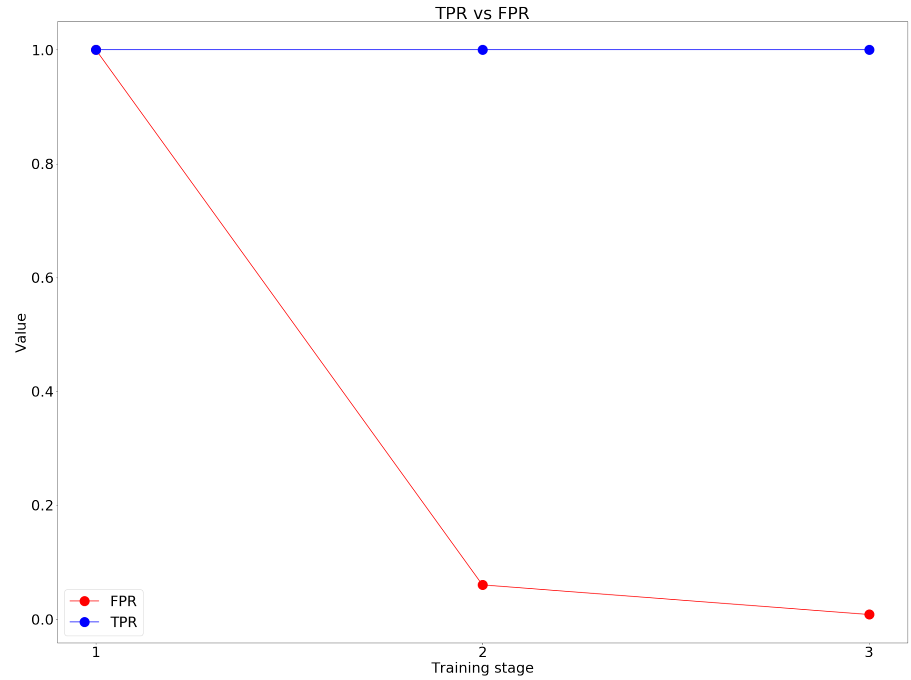

Figure 2: The true positive rate and false positive rate for each training stage.

The Viola-Jones classifier is a strong classifier constructed from a cascade of weak classifiers. Each stage in the classifier is constructed by training classifiers using AdaBoost[1]. AdaBoost is a supervised learning algorithm and so requires annotated positive and negative samples. The results shown in Figure 2 were generated using five hundred negative and five hundred positive samples, and the cascade was trained in three stages. At each stage the false positive rate decreases. However, the rate at which it decreases is reduced every time. This is because simpler classifiers are used to reject the majority of sub-windows before more complex classifiers are called upon to achieve low false positive rates[1] later on. If the cascade was trained using one thousand positive and negative samples, the false positive rates for stages two and three would be lower. The true positive rate stays at 100% throughout due to the `-minHitRate` argument of `opencv_traincascade` being 0.999.

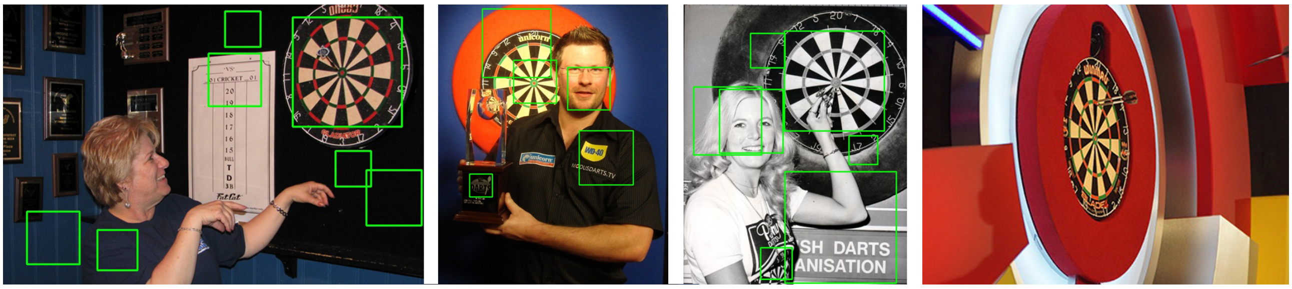

Figure 3: Dartboards detected by Viola-Jones classifier in `dart0.jpg`, `dart4.jpg`, `dart9.jpg`, and `dart12.jpg`.

Table 1: F_1 scores achieved for each image with the Viola-Jones classifier.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Average
0.25	0.33	0.00	0.29	0.00	0.14	0.00	0.18	0.10	0.18	0.00	0.20	0.00	0.17	0.05	0.67	0.16

The average true positive rate achieved by the classifier when classifying the test data set is 100%. Whereas the average true positive rate achieved when classifying the sixteen example dart images is 62.5%. This disparity in the values is due to the classifier being overfitted to the training data. The classifier was trained with variations of a perfect dartboard image created by `opencv_createsamples` and so performs extremely well on these images, but performs far worse on realistic images. In order to increase the performance on realistic dartboard images, a larger training set containing a greater variation of dartboard images could be used as this would result in less overfitting. It would also be a good idea to use negative images that are similar to the environments that dartboards are often found, rather than outdoor scenes used for the results above.

Third Subtask

Figure 4: The line hough space, gradient magnitude, and output for `dart11.jpg` (left) and `dart10.jpg` (right).

Table 2: Average precision, recall, and F_1 scores achieved by the Viola-Jones classifier and our classifier.

	Average precision	Average recall	Average F_1 score
Viola-Jones	0.097	0.625	0.160
Our classifier	0.531	0.531	0.521

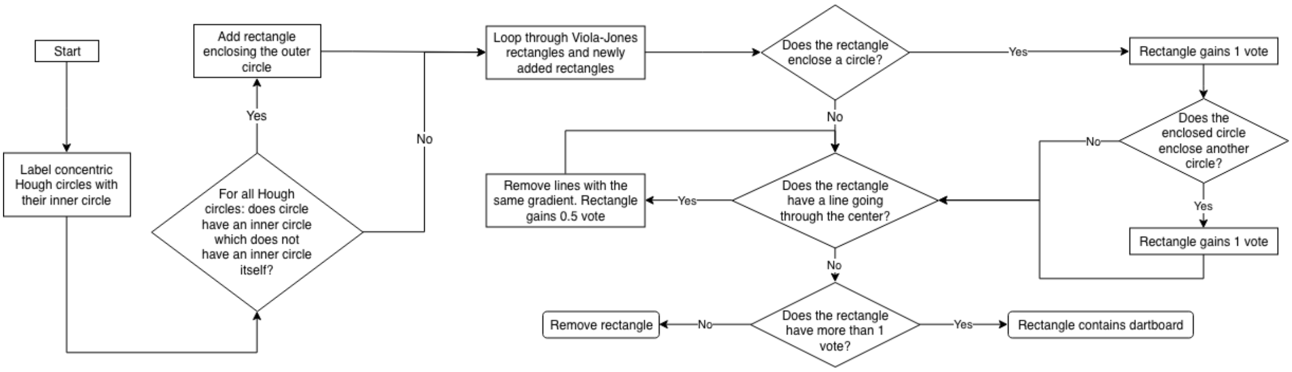

Figure 5: The combination of circles and lines detected by the Hough transform with the Viola Jones classifier.

Some key merits and shortcomings of our implementation:

- A merit of our system was that it recognized lines passing through the center of the potential dartboards that have unique gradients. This prevented multiple similar lines from skewing the number of votes that a rectangle received.
- A merit of our system was that it took into account both lines and circles provided by two separate Hough transforms.
- One shortcoming of our implementation was that it was not able to detect ellipses. This meant that if the image of the dartboard was not captured from straight on, our system had no way of detecting the dartboard unless there was an existing rectangle from the Viola-Jones classifier.

We combined the circles and lines identified by the Hough transform with the objects recognized by the Viola-Jones classifier in two stages. The first stage involved identifying additional dartboards that weren't detected by the Viola-Jones classifier. We felt there was a significant possibility of a dartboard being present when finding two concentric circles which shared a center location and whose ratios were 1:0.63. Upon finding these, we created a rectangle that enclosed the outer circle.

We now had a list of rectangles potentially enclosing dartboards. We created a 'voting' system that ranked these potential dartboards on their likelihood. The rectangles would gain votes for fulfilling the following criteria:

- If the rectangle enclosed a Hough circle, it received 1 vote. This represented the outer ring of the dartboard.
- Additionally, if the previous condition was fulfilled and there was another circle that could be deemed the inner circle or the previous circle (this is decided by comparing the ratio of the radii), it would receive another 1 vote.
- The rectangle would receive 0.5 votes for every line that passed near its center that had a unique gradient. This favoured rectangles that had many lines passing through at different angles.

After counting the votes for each potential dartboard, we discovered through testing that a sensible condition would be when a rectangle had at least 2 votes. For example, rectangles which either had two concentric circles or a circle with 2 lines of different gradients, would be allowed.

Fourth Subtask

Although we did not have time to implement any further object detection methods, we thought a lot about how it would be possible to do so:

- We attempted to train the classifier with more negatives and positives provided by the `opencv_createsamples` command and attempted to train the classifier over more stages, however this decreased the performance of the classifier with the example dartboard images provided as the classifier was simply even more overfitted as mentioned in the second subtask.
- We experimented with the options for training the cascade classifier, for example exploring the use of Local Binary Patterns (LBP) instead of the original Haar features. Initially, we thought the LBP features might allow for a better detection system for dartboards as they could better describe the dartboards 'texture' of black and white triangles. Additionally, the LBP cascade classifier was almost instantaneous to train, unlike the Haar cascade that took several minutes. However, after calculating the average F_1 scores for both of these classifiers, we discovered there was very little difference with LBP having an F_1 score of 0.145 compared to a score of 0.160 for Haar.
- In our attempt to improve the dartboard detector we looked at images that contained dartboards that our detector didn't identify, from this we found that it performed badly when the image was taken from large angles in respect to the dartboard. This meant that the they appeared in 2D as ellipses and hence were not detected by our system. Initially we attempted to implement ellipse detection using a Hough transform, where we would increment the Hough Space at (x, y, radius-y, radius-x). However, due to the hough space now existing in four dimensions, we quickly realized the algorithms intractability. We then did some research and contemplated implementing ellipse detection using an algorithm provided by the paper "A New Efficient Ellipse Detection Method"[2], however due to time constraints we didn't have time to implement this algorithm.
- Another weakness of our initial implementation meant that our generated Hough lines were infinite and weren't bounded on the image. Although we did not manage to detect the end of lines successfully due to gaps in the lines, if we had more time this could have a significant impact on the success of the detector. Bounding the lines would help as we could remove lines that were generated from other objects which coincidentally passed through the dartboard. It would also dramatically decrease the amount of line intersections found in the image, hence increasingly the likelihood that a intersection was in fact the center of a dartboard.
- Colour histograms could be used to help increase the accuracy of our system. The colour histogram of the original `dart.bmp` image could be normalised and then moved over the image (in a similar fashion to Haar features in the Viola-Jones classifier) and compared to the colour histogram of that area in the image. An algorithm such as the Earth Mover's Distance Algorithm[3] could be used for the comparison. If there was a similarity in the histograms, new rectangles could be generated for the rest of our system to vote on. This could also be used to vote for existing rectangles.

References

[1] P. Viola and M. J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, May 2004.

[2] Y. Xie and Q. Ji. A new efficient ellipse detection method. *International Conference on Pattern Recognition*, Vol. 2, 12 2002.

[3] Y. Xie and Q. Ji. A new efficient ellipse detection method. *International Conference on Pattern Recognition*, Vol. 2, 12 2002.