COMS30030: Image Processing and Computer Vision
Coursework Report

Tom Stark (jb18789)

1 Viola-Jones Object Detector

Ground Truth and Visualisation

Ground truths help to visualise the true values of an object within an image, In this first example there are red bounding boxes displaying the ground truth values for the faces within the images and the green bounding boxes show the faces detected by the Viola-Jones detector.

Fig. 1: `dart4.jpg`

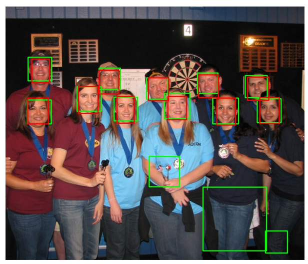


Fig. 2: `dart5.jpg`

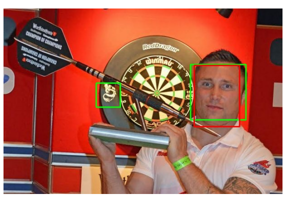


Fig. 3: `dart13.jpg`

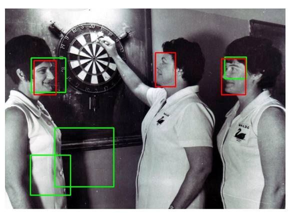


Fig. 4: `dart15.jpg`

Fig. 5: `dart14.jpg`

Fig 4 shows the extent to which I chose Ground Truth values for 'frontal' faces. Images produced with `face_truth.cpp`.

IOU, TPR, F1-Score

The Intersection Over Union (**IOU**) is an evaluation metric used to measure the accuracy of an object detector. In this case we are evaluating the relative overlap of the ground truth bounding boxes with the predicted bounding boxes from the detector.

$$IOU(r_1, r_2) = \frac{r_1 \cap r_2}{r_1 \cup r_2}$$

Where r_1 and r_2 are rectangles. $r_1 \cap r_2$ is the area of intersection and $r_1 \cup r_2$ is the area of union.

The True Positive Rate (**TPR**) is number of correct detections over the number of total truths for an image. 5 of the 16 test images did not include any frontal faces so the TPR did not yield any relevant results. While the TPR is a useful measure of the performance of a detector it may be difficult to assess the meaningfulness of the TPR if the test image does not contain the object you are trying to detect.

It is always possible to achieve a TPR of 100% since the IOU threshold is manually set. If you have a low threshold the rate of a positive detection naturally increases. I found a value of 0.4 to give good results.

Another measurement to determine the effectiveness of a detector is the Positive Predictive Value (**PPV**), calculated by the number of true positives over the sum of the total predictions. This gives you an idea of the precision of your detector.

The PPV is used with the TPR to work out the F_1 score, which is a measurement of the accuracy of the detector. It can be calculated using this formula

$$F_1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)}.$$

Where TP is the true positives, FP the false positives and FN the false negatives.

I used a manually fixed IOU threshold of 0.4 to give the following results:

Test File	True Positive Rate	F1-Score
`dart0.jpg`	0.000	0.000
`dart1.jpg`	NaN	NaN
`dart2.jpg`	NaN	NaN
`dart3.jpg`	NaN	NaN
`dart4.jpg`	1.000	1.000
`dart5.jpg`	1.000	0.880
`dart6.jpg`	0.000	0.000
`dart7.jpg`	1.000	1.000
`dart8.jpg`	0.000	0.000
`dart9.jpg`	1.000	0.400
`dart10.jpg`	NaN	NaN
`dart11.jpg`	1.000	1.000
`dart12.jpg`	NaN	NaN
`dart13.jpg`	1.000	0.667
`dart14.jpg`	1.000	0.500
`dart15.jpg`	0.667	0.571
Average	0.697	0.547

Pictures without visible faces have been given a value of 'NaN' to remove unwanted skew in the results.

I previously had the IOU threshold at 0.5, which meant that 50% of the area of the detected bounding box had to overlap with the ground truth bounding boxes. I decided to reduce this to 0.4 since it gave slightly better results.

2 Building & Testing a Detector

Training Performance

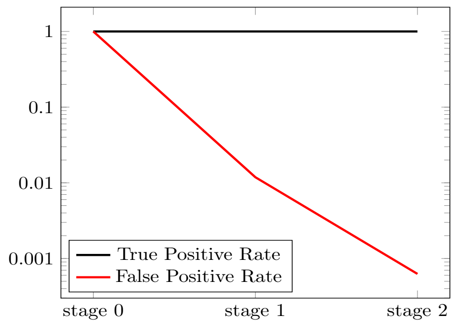


Fig. 6: TPR vs FPR during the dart classifier training.

The False Positive Rate (**FPR**) is the fraction of false positives over the sum of the number of false positives and true negatives. It is another way to determine the accuracy of a detector.

During the training the TPR remained at 100% for each of the stages, while the FPR shows an exponential drop-off throughout the training. This tells us that the probability of the trained classifier producing false positive detections in relation to the number of negative events is 0.00623%.

Testing Performance

These images are a selection of images that were produced from running the Viola-Jones detector with the new trained classifier on the test images. The green bounding boxes show the detected dartboards and the red bounding boxes show the ground truths.

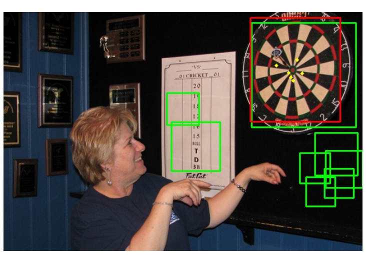


Fig. 7: `dart4.jpg`

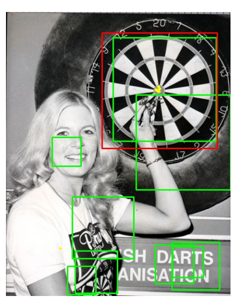


Fig. 8: `dart5.jpg`

Fig. 9: `dart14.jpg`
Images produced with `dart_truth.cpp`, using only Viola-Jones.

The TPR and F_1 score for each image is shown in the table below. I used a manually fixed IOU threshold of 0.4.

Test File	True Positive Rate	F1-Score
`dart0.jpg`	1.000	0.222
`dart1.jpg`	1.000	0.667
`dart2.jpg`	0.000	0.000
`dart3.jpg`	1.000	0.333
`dart4.jpg`	0.000	0.000
`dart5.jpg`	1.000	0.182
`dart6.jpg`	0.000	0.000
`dart7.jpg`	0.000	0.000
`dart8.jpg`	1.000	0.267
`dart9.jpg`	1.000	0.200
`dart10.jpg`	1.000	0.286
`dart11.jpg`	0.000	0.000
`dart12.jpg`	1.000	0.667
`dart13.jpg`	0.000	0.000
`dart14.jpg`	1.000	0.143
`dart15.jpg`	1.000	0.667
Average	0.625	0.227

These results show the detector can correctly identify an average of 62.5% of dartboards in an image. Compared to the face detector the TPR value of the trained dart classifier is 10.3% lower. Therefore, I concluded that it has a lower average detection rate. This is likely due to the fact that, faces are easier to detect because they have more distinctive features than a dartboard. Faces are also usually the main focus of an image and therefore detecting a dartboard as part of the background can be more difficult.

Although the classifier was trained with different viewing angles, its main strength is detecting dartboards from an image where a picture was taken with the dartboard facing the camera and not at an angle where it can become an elliptical shape. This is another reason why the detector is not as effective.

In addition, the classifier was trained on sample images taken from a single picture of a dartboard. In reality, a dartboard may not always look the same, or have different angles, or different contrast from the dartboard it was trained on.

The F_1 score for the dartboard detector also decreased with a value of -0.320. This was a much more significant drop than the TPR. This means the detector is less accurate, most likely due to the fact that it produces more false positives than the face detector.

The classifier trained for dartboards with the Viola-Jones detector was not very reliable in detecting dartboards. This presents a problem when you take a filtering approach based on only removing bounding boxes that are already present without adding additional bounding boxes. Therefore while the TPR or F_1 score may increase, this approach to filtering can only be as good as the initial classifier.

3 Shape Detector Integration

Hough Details

The images below outline how the hough space for both circles and lines can be used to filter bounding boxes. My HoughTransform produces the hough spaces depicted in the pictures below.

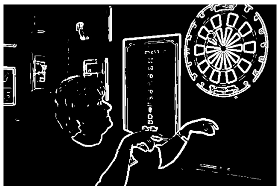


Fig. 10: `dart0.jpg` Magnitude threshold.

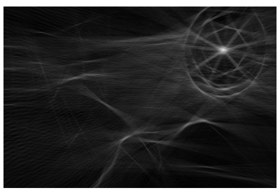


Fig. 11: `dart0.jpg` Hough Space lines


Fig. 12: `dart11.jpg` Magnitude threshold.

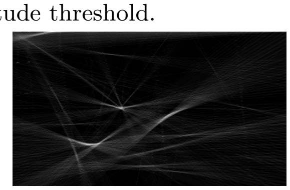


Fig. 13: `dart11.jpg` Hough Space lines.

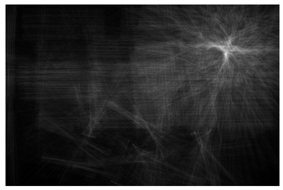


Fig. 14: `dart0.jpg` Hough Space circles.


Fig. 15: `dart0.jpg` Filtered bounding boxes.

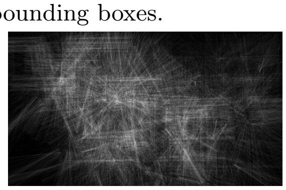


Fig. 16: `dart11.jpg` Hough Space circles.


Fig. 17: `dart11.jpg` Filtered bounding boxes.

Evaluation

The TPR and F_1 score for each image is shown in the table below, together with the differences from the previous version of the detector. I used a manually fixed IOU threshold of 0.4

Test File	TPR	TPR Diff.	F_1	F_1 Diff.
`dart0.jpg`	1.000	±0.000	1.000	+0.778
`dart1.jpg`	1.000	±0.000	0.667	±0.000
`dart2.jpg`	0.000	±0.000	0.000	±0.000
`dart3.jpg`	1.000	±0.000	0.667	+0.333
`dart4.jpg`	0.000	±0.000	0.000	±0.000
`dart5.jpg`	1.000	±0.000	0.400	±0.218
`dart6.jpg`	0.000	±0.000	0.000	±0.000
`dart7.jpg`	0.000	±0.000	0.000	±0.000
`dart8.jpg`	0.500	−0.500	0.500	+0.233
`dart9.jpg`	1.000	±0.000	1.000	±0.800
`dart10.jpg`	0.667	−0.333	0.667	±0.381
`dart11.jpg`	0.000	±0.000	0.000	±0.000
`dart12.jpg`	1.000	±0.000	0.667	±0.000
`dart13.jpg`	0.000	±0.000	0.000	±0.000
`dart14.jpg`	0.500	−0.500	0.500	+0.357
`dart15.jpg`	1.000	±0.000	0.667	±0.000
Average	0.542	−0.083	0.421	+0.194

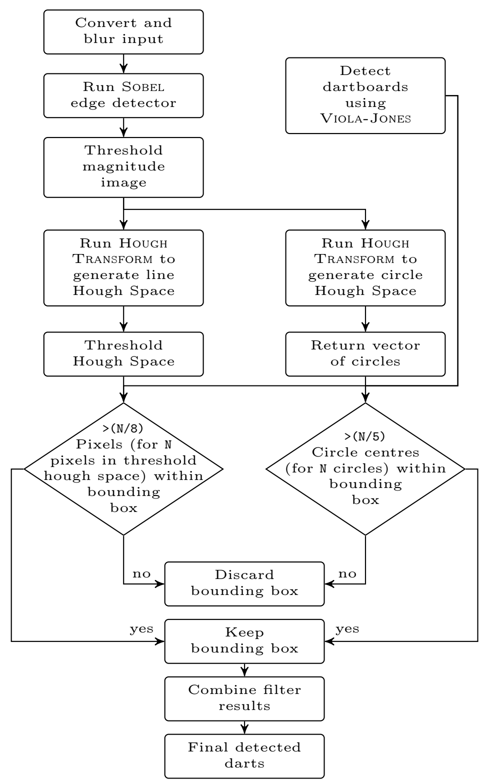
The benefits of the enhanced implementation:

- The precision of the detector increases as lots of obvious false positives are filtered out.
- Increased F_1 Score, increasing the overall accuracy of the detector.

The disadvantages of the enhanced detection method:

- The TPR shows a slight decrease, due to some images containing dartboards of an elliptical shape, or where an image lacks contrast or brightness to be detected by either hough space e.g. `dart11.jpg`.
- False positives still remain, so further filtering would be required.
- New bounding boxes cannot be created based on the hough space, only existing boxes are filtered out.

Detection Pipeline



The reasoning behind this method of detection:

- Bounding boxes that did not contain any line pixels or circle centres were unlikely to be a dartboard.
- I didn't want to set a fixed threshold for allowing a bounding box, therefore a value of $n/8$ and $n/5$ for pixels and circle centres respectively gave good results.
- I did not remove overlapping boxes. Although this gives a decreased F_1 score, it was important for the further improvements.

4 Improving the Detector

Idea

The idea was to improve the TPR and F_1 score by combining overlapping bounding boxes that had all correctly identified a dartboard. I saw a common pattern where there were multiple bounding boxes over circle centres or lines that were either a lot smaller or a lot larger than the ground truth. This meant the IOU was too low to be a successful detection for any of the bounding boxes, and the F_1 score decreased due to the overlap.

If you combine all of these bounding boxes and average out their x, y coordinates and their width and height, you should be left with a single bounding box that has a much better fit wrt. the ground truth. Therefore, improving both the TPR and the F_1 score.

I created a rectangle grouping function GroupDarts that takes in the filtered `vector<Rect>` produced by the hough space filtering method, and returns a new `vector<Rect>` of grouped bounding boxes.

The function checks the new vector to see if a rectangle from the old vector intersects with it.

- If it does, the rectangle is averaged with the one intersecting it, and is removed from the old vector.
- If it doesn't, it is added to the new vector.

It was important that this function worked correctly so that it didn't combine bounding boxes when there were multiple dartboards in the image.

Visualise

These images show the bounding boxes before and after the GroupDarts function

Fig. 18: `dart14.jpg` Predictions before grouping.

Fig. 20: `dart14.jpg` Predictions after grouping.

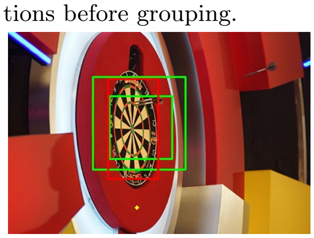


Fig. 19: `dart12.jpg` Predictions before grouping.

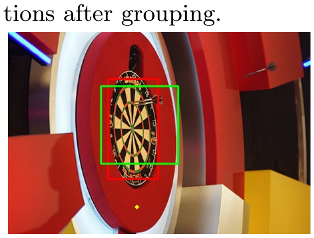


Fig. 21: `dart12.jpg` Predictions after grouping.

Images produced with `dart_truth.cpp`, using HoughTransform and grouping.

Evaluation

The average TPR and F_1 score is shown in the table below, together with the difference in averages to the previous versions of the detector. I used a manually fixed IOU threshold of 0.4

Test File	TPR	F_1 Score
Average	0.510	0.488
Diff. to original implementation	−0.177	+0.219
Diff. to hough space implementation	−0.094	+0.025

The F_1 score increased by 50% over the original implementation.

The benefits of the final implementation:

- Increased F_1 Score, increasing the overall accuracy of the detector.
- Successfully detects over 50% of dartboards in all images.
- Very few false positive detections.

The disadvantages of the final implementation:

- The TPR shows a slight decrease, due to some overlapping bounding boxes being much larger than the boxes it was averaged with. This resulted in an averaged bounding box with an IOU under the threshold. Therefore bounding boxes that were detected before are now no longer detected.
- Some circles were detected in places where there are no dartboards, resulting in some false positives, especially when a high percentage of the circles were detected there.
- Elliptical shapes still present a problem especially if there are also no lines detected in this area.

In addition to grouping the bounding boxes, I implemented a filter based on the average radius of the circles detected in an image. It filtered out bounding boxes where the width or height was below half the average radius or above double the diameter. This solved most issues with over or undersized bounding boxes that would skew the grouping, resulting in an increased TPR. However, it did not take into account that circles detected within an image are not necessarily from the dartboard. With some images there were large circles detected with a radius double the diameter of the dartboard, which subsequently negated the TPR increase. Therefore I decided against including this approach.

Choosing where to place ground truth boxes also produces varying results since choosing to include the black outer ring can result in an increased TPR if your detected bounding boxes tend to be larger than the dartboard.

Overall, the final implementation does a good job at detecting dartboards within an image and a better job than the original implementation. Nevertheless there are still lots of problems with this implementation and additional improvements that can be made.