COMS30121 - Image Processing and Computer Vision
The Dartboard Challenge

Joshua Van Leeuwen & Karim Allaouat

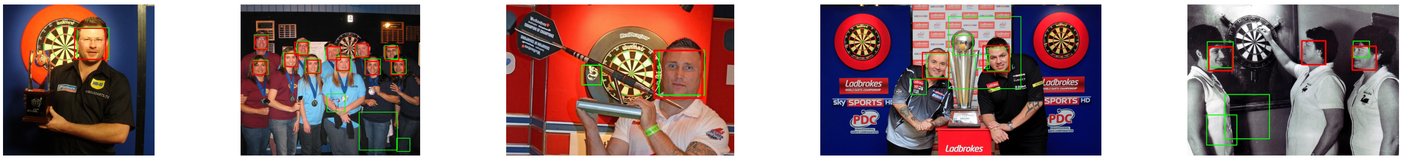
Introduction

This task introduces the ability to detect and locate instances of an object class in images. This is important as this ability is used in many computer vision applications. The task explores the Viola-Jones object detection framework (an "off the shelf" face detector) and combines it with other detection techniques to improve it. The image set used is from the popular sport, darts.

The Viola-Jones Object Detector

The Viola-Jones object detection framework is the first object detection framework to provide competitive object detection rates in real time. The algorithm was used with a strong classifier trained using AdaBoost for detecting human faces from the front.

Using the Detector on Human Faces



(a) dart4.jpg. (b) dart5.jpg (c) dart13.jpg (d) dart14.jpg (e) dart15.jpg

Figure 1: Result of the Viola-Jones Algorithm on human faces (green boxes). Red boxes represent ground truth.

Assessing How the Detector Performs

The TPR or True Positive Rate measures the proportion of relevant items that are correctly identified. In this case it is the fraction of successfully detected faces out of all valid faces in an image. The TPR of dart5.jpg and dart15.jpg are 100% and 67% respectfully.

A practical difficulty of computing the TPR accurately is that the hits and misses have to be manually counted. Also, errors can occur when faces are side profile because they become ambiguous as to whether they are valid. It is always possible to get 100% TRP because you can detect everything in the image and so will always get all possible hits. It will however, get all the misses too. A better way of evaluating the detector would be to calculate the F_1 score. It takes into account the detectors precision (PPV - Positive Prediction Value, how many selected items are relevant) and recall (TPR). A set of rules were created to evaluate whether a face was valid:

- Two eyes and a mouth must be within a boundary to be counted as a hit.
- Two eyes and a mouth must be visible to us in order for it to be counted as valid.
- The F_1 score will be calculated by:

$$\frac{2 \times P \times R}{R + P}$$

  Where

- Recall (TPR) $P = \frac{truepositives}{groundtruth}$.
- Precision $R = \frac{truepositives}{truepositives + falsepositives}$.

As calculating the F1 score is challenging due to manually counting boxes, a process was implemented that makes this easier and scalable. It will compare the centres of the ground truth (which are manually added) and detection boxes and if they are below a certain threshold, will count as a true positive. Table 1 below shows the result of this.

Picture	Actual	Detected	Hit	Missed	F_1 Score
dart4	1	1	1	0	1
dart5	11	14	11	3	0.88
dart13	1	2	1	1	0.67
dart14	2	6	2	4	0.5
dart15	3	4	2	2	0.5

Table 1: Comparing the F_1 Score of different images.

Building and Testing the Detector

Interpreting TPR vs FPR

Figure 2 shows the training of the detector over the 3 stages. The TPR always remained as 1 therefore, it was successful in detecting all dartboards. The decreasing FPR portrays that the detector firstly detects as much as it can, then reduces the number of objects it detects. As a consequence, it is clear that the detector is improving. The parameters of the detector were changed to be optimum. A ratio of 500:1000, positive to negative was used and a max false alarm rate of 0.4. TODO: explain parameter tuning.

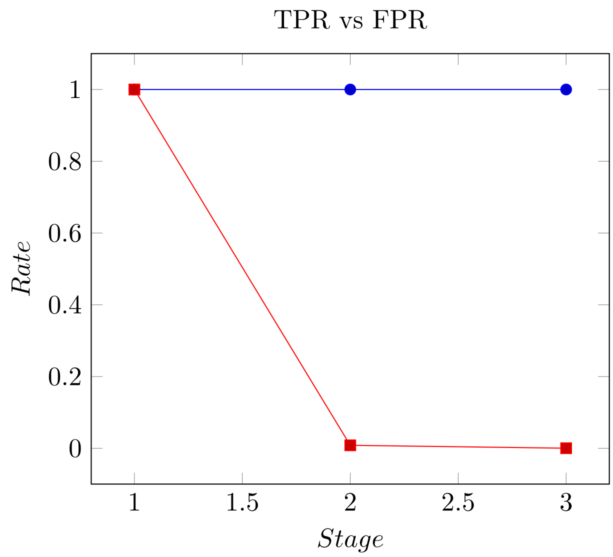


Figure 2: TPR (blue) vs FPR (red) across the 3 stages.

Testing on images

(a) dart4.jpg. (b) dart5.jpg (c) dart13.jpg (d) dart14.jpg

Figure 3: Result of the trained dartboard detector.

TODO: make this look beter?
The F_1 of the images are:

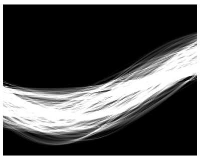
- dart0.jpg - 0.14.
- dart1.jpg - 0.13.
- dart2.jpg - 0.12.
- dart3.jpg - 0.20.
- dart4.jpg - 0.20.
- dart5.jpg - 0.10.
- dart6.jpg - 0.17.
- dart7.jpg - 0.09.
- dart8.jpg - 0.13.
- dart9.jpg - 0.13.
- dart10.jpg - 0.11.
- dart11.jpg - 0.29.
- dart12.jpg - 0.33.
- dart13.jpg - 0.14.
- dart14.jpg - 0.07.
- dart15.jpg - 0.25.

The overall F_1 score is consequently 0.1625. The F_1 score is relatively low therefore the denominator is much bigger and can conclude that there was a high number of detections with respect to hits. This means that there were a lot of misses. The usefulness of the plot (Figure 2) is that it can be clearly seen that the detector is currently under fitting as the TPR remains at 1 and the FPR decreases. This fact, along with the F_1 scores, portray the results of a poor detector. However, it can be used to an advantage. The under fitting can be combined with other classifying detectors in order to improve results.

Integration with Shape Detectors

Image Results

(a) Threshold Gradient Magnitude.

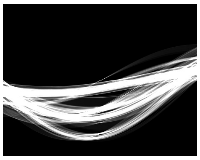


(b) Hough Space

(c) Result

Figure 4: dart5.jpg shows the merits of the detector.

(a) Threshold Gradient Magnitude.



(b) Hough Space

(c) Result

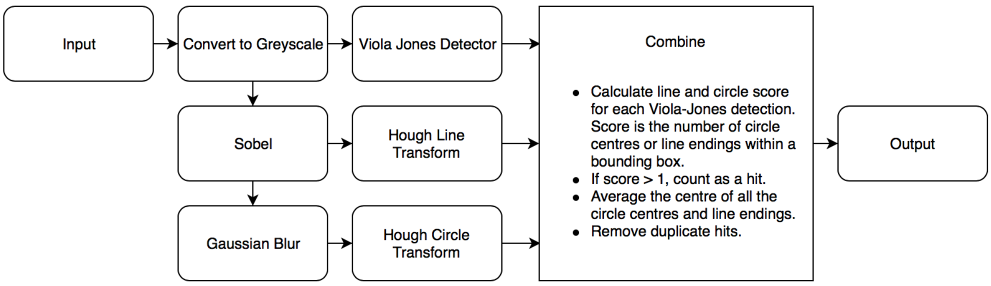
Figure 5: dart10.jpg shows the limitations of the detector.

The new dartboard detector did considerably better than the previous, achieving an overall F_1 score 0.767 with the previous being 0.163.

Merits and Limitations

- This detector adds more classifiers when analysing images meaning that the large set of detections with many negative hits, is able to be reduced by combining each classifier result.
- This detector works optimally with images where dartboards are in good lighting and are facing straight at the camera in the scene.
- As shown in the dart10.jpg image, the detector can fail to detect two dartboards that are at an angle.
- These failures are due to the circle and line Hough transformation being used in which struggle to detect non-uniform shapes.

Combination of Detectors



(a) Flowchart representing the detector.

- Our approach was to create new classifiers which would refine the large number of negative and positive hits of the Viola Jones detector hits down to only true positives by accepting Viola Jones hits that were also observed by the line and circle detectors.
- Line detections are acheived by acepting a hit when a large number of lines intersect at a pixel position, reletive with the number of lines found in the image.
- This involves iterating through the set of Viola hits, comparing whether any line or circle hits are contained in the Viola bounding box, accepting if so, rejecting otherwise.
- Accepted hits would change its location based on the average position of itself, along with its combined detections.

- Circle detections also give the ability to estimate the size of the dartboard. Furthermore, it takes the average radii of included circles and includes this in the approximation.
- To reduce duplicate detections of the same dartboard, overlapping bounding boxes are combined, also averaging location and size - ensuring one positive hit per dartboard.

Improving the Detector

- In order to improve our dartboard classifier, we considered further shapes which help to classify a dartboard being present. As such, we identified two shapes - ellipses and triangles. Firstly, we realised that some dartboards had been not been identified by our classifier due to not being detected by circles with dartboards that are at an angle. By using ellipses, we would be able to capture this shape of valid dartboards. We also chose to consider triangle shapes inside images to detect dartboards as all dartboards have distinct triangles contained that will give more detections that can be combined in our detector.
- Detecting both triangle and ellipse shapes is achieved by first applying the Canny edge detector on the grey scale input image. The Canny edge detector, developed by John F. Canny in 1986 [1] works by calculating the gradient magnitude of each pixel. To determine whether each pixel is part of an edge, two thresholds are applied where pixels greater than the high threshold are considered pixels on an edge, pixels below the low threshold are discarded and pixels in-between the two thresholds are considered edges only if they are connected to a pixel that is above the higher threshold. In order to select an appropriate threshold for each image, the Otsu method is applied. [2]
- The Otsu method is used to determine the largest threshold input for the Canny edge detector and it assumes that the image contains two classes of pixels. It then calculates the optimum threshold and partitions the two classes so that their combined variance is maximal. We selected out minimal threshold to be one third of the maximum gradient the Otsu method returned.
- With our Canny image we then used the *findCountours* function which attempts to find groups of points which together form a curve. With these contours we can then determine whether, in the case of triangles, form a closed shape of 3 sides, or in the case of ellipses, a contour group with many elements is indicative of a many sided shape - an ellipse.
- In combination with the new shapes being in included into our detector, we also investigated another type of image processing technique called *Speeded Up Robust Features* (SURF).
- TODO: Surf Method. We use a scoring system with surf. We cluster surfs together that count as the same object.
- TODO: find better viola jones parameters.
- Include images to show off improvements.
- After implementing this the F1 score became 0.8666667333333333333333.

References

1. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.420.3300&rep=rep1&type=pdf
2. https://en.wikipedia.org/wiki/Otsu%27s_method