COMS30121 - Dartboard Challenge

Luke Storry (LS14172) Alex Robinson (AR15247)

November 30, 2017

Subtask 1

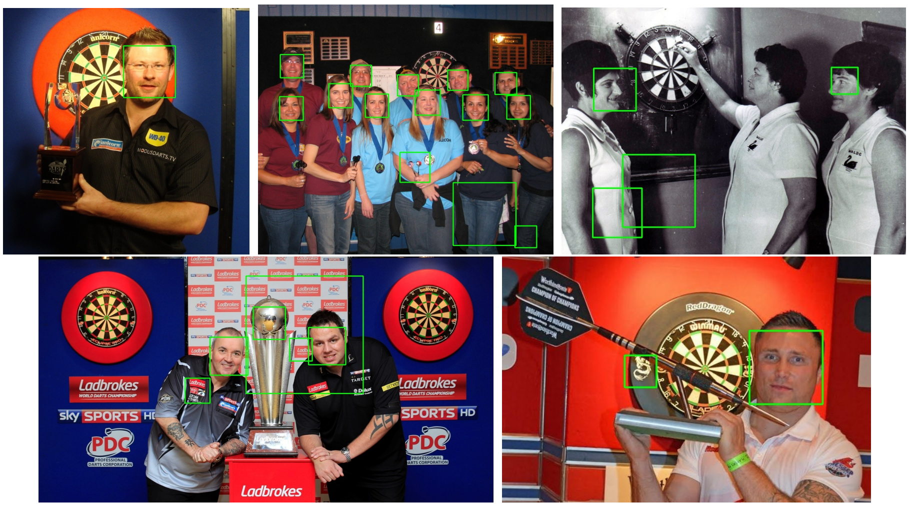

Figure 1: Viola-Jones Face Detection[14] on five example images: `dart4.jpg`, `dart5.jpg`, `dart13.jpg`, `dart14.jpg` and `dart15.jpg`

For our ground truth we considered frontal faces as valid faces and annotated the the images to accept a true positive if the bounding box contained the eyes and mouth. The true positive rate (TPR) for **dart5.jpg** is $11/11 = 1.00$ as all faces were detected, however it also detected two false negatives. For **dart15.jpg** four false positives were detected however the image contained no valid (frontal) faces and therefore had a TPR of 1.00.

The main practical difficulty in assessing the TPR is that a large variety of different images are required for an accurate result. Manual testing is too time-consuming, but an accurate automated test still requires manual annotations of the images with ground truths, which can be very time-consuming for large datasets. The level of detail required for a good ground-truth can be higher than expected: simply counting the number of faces is rarely sufficient, as the placement could be wrong, so ground-truth annotation should also include coordinates of where the faces are, which also leads to the issue of how close the detector would have to be to count it as a 'positive'.

It is always possible to achieve a 1.00 TPR in any detection task, by simply 'detecting' every set of pixels as a positive. This is obviously an absurd situation, but over-sensitivity can be an issue in developing image-detectors, so combining the TPR with the false positive rate (FPR), and using the F1 score, is a much more realistic metric.

One set of measures and rules to accurately calculate the $\texttt{F}_1$ score for a face detector could be to annotate the images with a list of optimal bounding box coordinates, and accept the detected boxes as a positive match if their coordinates are within a given proximity of the ground truth coordinates. For example for the above image `dart14.jpg`, the ground truth coordinates would be:

$$[[(458, 212), (562, 314)], [(719, 180), (824, 290)]]$$

and we would mark a detected box as a true positive if its coordinates were within 5px either side of that ground truth. Any incorrect boxes or missing boxes would then be labeled as false positives and false negatives respectively.

Subtask 2

After the first stage of training the classifier, both the reported TPR and FPR are one, which signifies that the Viola-Jones classifier[13] would detect far too many dartboards, both true and false. Then at each successive stage of the classifier, the reported FPR decreases while the TPR maintains at one, as seen in Figure 2. This means that the classifier gets better at removing false positives while maintaining the true positives in the generated training samples.

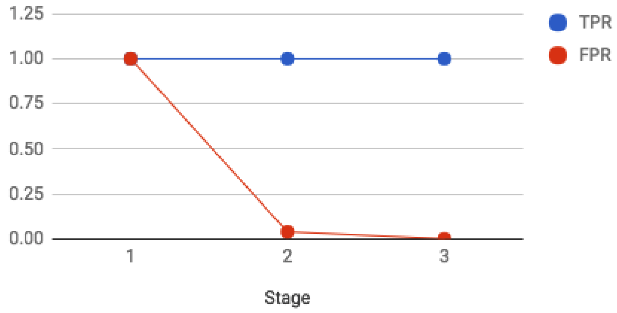

Figure 2: Reported TPR by FPR for each stage of Viola-Jones classification on generated sample training images

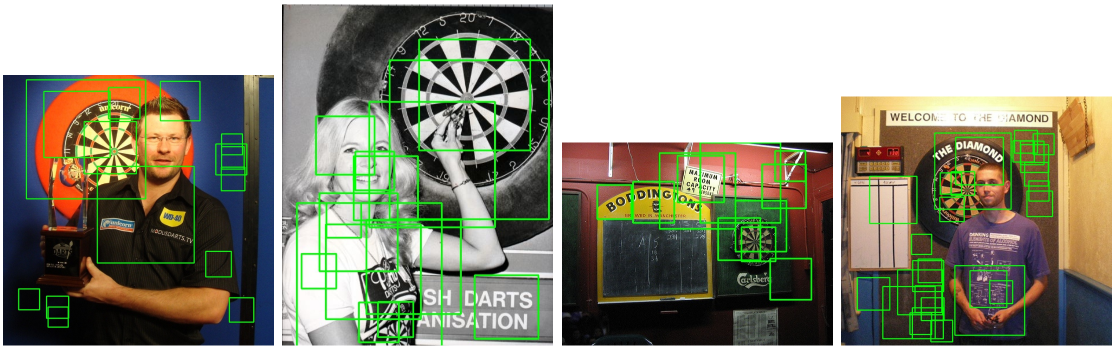

Figure 3: Dartboard detection

$$\begin{aligned}
\text{precision} &= \frac{TP}{TP+FP} && = \frac{20}{20+7654} && = 0.0026 \\
\text{recall} &= \frac{TP}{TP+FN} && = \frac{20}{20+0} && = 1.0000 \\
\mathrm{F}_1 &= 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} && = 2 \cdot \frac{0.0026 \cdot 1.0000}{0.0026 + 1.0000} && = 0.0052
\end{aligned}$$

This $\mathtt{F}_1$ score is very poor, which shows that the plot in Figure 2 is a poor measure of predicting system performance: although it tells us we have a TPR of and very low FPR on the training images, it doesn't give any strong indication on how precise the model is on other test images.

Subtask 3

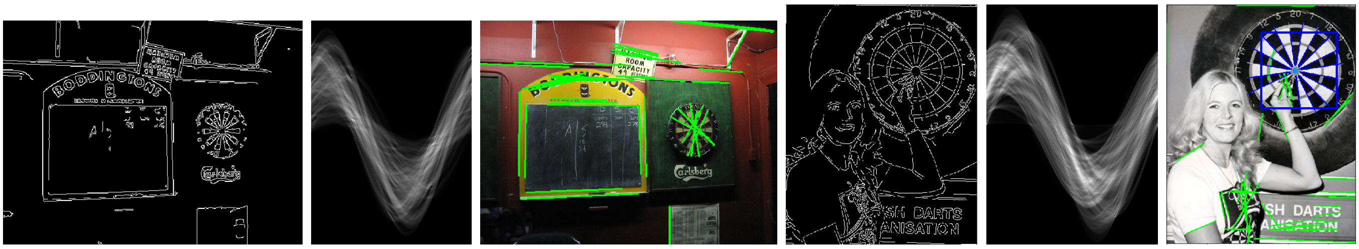

Figure 4: Thresholded gradient magnitudes[2], 2D representations of the Hough Space, and the final detection images, for dart3.jpg and dart9.jpg

After tuning for the highest $\mathtt{F}_1$ score, our detection metrics are:

$\mathtt{F}_1$: 0.44, precision: 0.58, recall: 0.35

One key merit of our detector is that it can accurately detect the centrepoint and scoringlines of dartboards, even when partially obscured. One shortcoming is that, to make the $\mathtt{F}_1$ score as high as possible, we tuned the detector to be very discriminative, thus removing many false positives (and increasing precision) but with the drawback of a low recall. For example see the third image in Figure 4, where the dartboard in `dart3.jpg` is not detected because the Hough-detected lines do not all intersect in the same place.

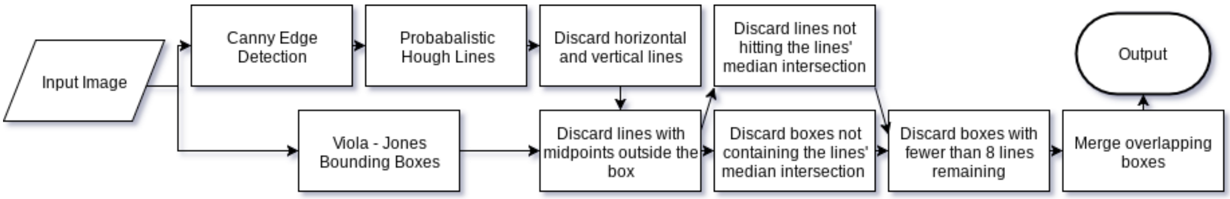
Input Image
Canny Edge Detection
Probabalistic Hough Lines
Discard horizontal and vertical lines
Discard lines not hitting the lines' median intersection
Output
Viola - Jones Bounding Boxes
Discard lines with midpoints outside the box
Discard boxes not containing the lines' median intersection
Discard boxes with fewer than 8 lines remaining
Merge overlapping boxes

Figure 5: Flow Diagram depicting combination of evidence

The Viola-Jones object detection from Subtask 2 could detect almost all dartboards, but had many false positive bounding boxes on every image. Thus we used the evidence from a Probabalistic Hough Lines detector [6] to loop through each potential dartboard and discard it based on whether enough intersecting lines were in the bounding box. We attempted to detect circles with another Hough transform[15], and use concentric circles as further evidence, but as the hough line detection was sufficient on all non-noisy dartboards, using the evidence from circles just added more false negatives without providing any true positives that weren't already detected by the lines. We also experimented with elipse-detection[12], but due to the noise of the images and the larger parameter spce drastically increasing time, we found that this overall did not help our detector either.

Subtask 4

To further improve our detector we needed to improve the recall without drastically reducing the precision, which the method from Task 3 struggled with despite copious tuning - i.e. we needed to detect more true positives without false positives.

After trialing the use of intensity histograms, our final approach combined scale-invariant template matching[1], which selects a bounding box with the highest cross-correlation coefficient [3], and then calculating SIFT[5] features within the box and finally using a FLANN-based matcher[9][8] to match features with the query image.

Figure 6 shows two examples from our detector. Scale-invariant template matching was achieved by with multiple passes, re-scaling our template image appropriately. Only matching SIFT features from the best template matching bounding box dramatically reduced the noise encountered when using the entire image or using the bounding boxes returned by Viola Jones. This helped to increase our recall as we could now detect previously undetected images, and as it produced no false positives had no negative impact when combined with our task 3 detector.

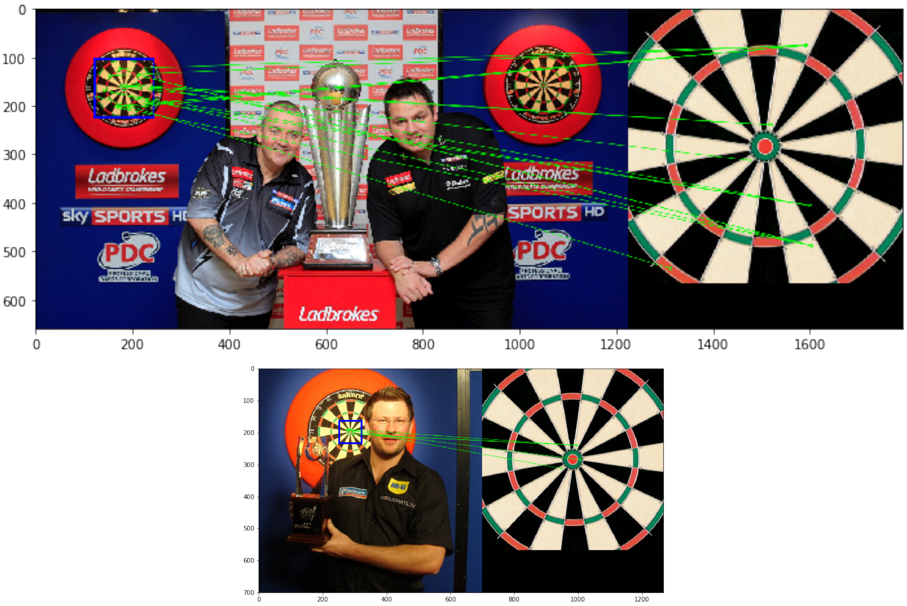

Figure 6: Template matching with SIFT features

Our final detector combined the method described above with the method from task 3 and produced final scores of: F_1= 0.65, precision = 0.71, recall = 0.60. An improvement to our F_1 score of 0.22 over our task 3 detector.

Evaluation of final detectors merits and shortcomings:

- Great at detecting previously missed dartboards which weren't detected because the edge detection was either too noisy or too sparse.
- We didn't transform our template image and therefore our detector was poor at dealing with skewed dartboards. An extension would be to generate multiple transformations and apply these however this would increase detection time.
- Poor at dealing with multiple dartboards in a single image, currently only detects the best candidate, so could be extended with multiple passes.

References

[1] K. Briechle and U. D. Hanebeck. “Template Matching Using Fast Normalized Cross Correlation”. In: *Proceedings of SPIE: Optical Pattern Recognition XII.* Vol. 4387. Mar. 2001, pp. 95–102. DOI: 10.1117/12.421129. URL: http://dx.doi.org/10.1117/12.421129.

[2] J. Canny. “A Computational Approach to Edge Detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6 (Nov. 1986), pp. 679–698. ISSN: 0162-8828. DOI: 10.1109/TPAMI.1986.4767851.

[3] J. P. Lewis. *Fast Normalized Cross-Correlation.* 1995.

[4] Rainer Lienhart and Jochen Maydt. “An Extended Set of Haar-Like Features for Rapid Object Detection”. In: *IEEE ICIP 2002.* 2002, pp. 900–903.

[5] David G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *Int. J. Comput. Vision* 60.2 (Nov. 2004), pp. 91–110. ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000029664.99615.94. URL: https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[6] J. Matas, C. Galambos, and J. Kittler. “Robust Detection of Lines Using the Progressive Probabilistic Hough Transform”. In: *Comput. Vis. Image Underst.* 78.1 (Apr. 2000), pp. 119–137. ISSN: 1077-3142. DOI: 10.1006/cviu.1999.0831. URL: http://dx.doi.org/10.1006/cviu.1999.0831.

[7] J. Matasyx et al. “Progressive Probabilistic Hough Transform”. In: 1998.

[8] Marius Muja. *FLANN, Fast Library for Approximate Nearest Neighbors.* https://www.cs.ubc.ca/research/flann/. 2008.

[9] Marius Muja and David G. Lowe. “Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration”. In: *International Conference on Computer Vision Theory and Application VISSAPP'09).* INSTICC Press, 2013, pp. 331–340.

[10] OpenCV. *Open Source Computer Vision Library.* https://github.com/opencv/opencv. 2015.

[11] *The OpenCV Reference Manual.* 2.4.9.0. Itseez. Apr. 2014.

[12] S. Tsuji and F. Matsumoto. “Detection of Ellipses by a Modified Hough Transformation”. In: *IEEE Trans. Comput.* 27.8 (Aug. 1978), pp. 777–781. ISSN: 0018-9340. DOI: 10.1109/TC.1978.1675191. URL: http://dx.doi.org/10.1109/TC.1978.1675191.

[13] Paul Viola and Michael Jones. “Robust Real-time Object Detection”. In: *International Journal of Computer Vision.* 2001.

[14] Paul Viola and Michael J. Jones. “Robust Real-Time Face Detection”. In: *Int. J. Comput. Vision* 57.2 (May 2004), pp. 137–154. ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000013087.49260.fb. URL: https://doi.org/10.1023/B:VISI.0000013087.49260.fb.

[15] H. K. Yuen et al. “Comparative Study of Hough Transform Methods for Circle Finding”. In: *Image Vision Comput.* (). DOI: 10.1016/0262-8856(90)90059-E. URL: http://dx.doi.org/10.1016/0262-8856(90)90059-E.