Image Processing and Computer Vision
Dartboard Detection Coursework Report

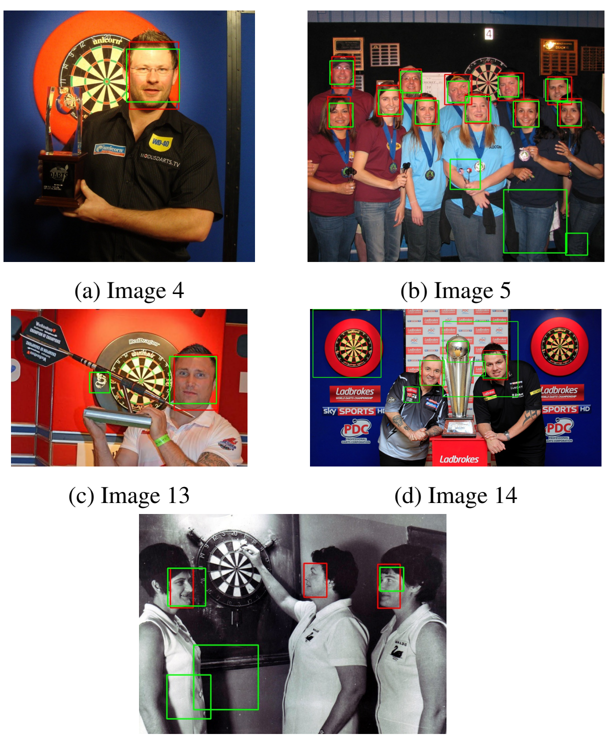
Zack Dove, Jake Ramaer

Introduction

In this project we will be exploring, implementing and analysing image processing and computer vision techniques in the context of identifying dartboards in images. We will experiment with the Viola-Jones object detection framework[1], and combine it with other techniques to improve it's efficacy. We will be working in C++ and OpenCV.

1 The Viola-Jones Object Detector

Firstly, we used the provided `frontalface.xml`, which contains a strong classifier trained using AdaBoost for detecting frontal human faces, with the Viola-Jones framework to build a detector for faces.



(a) Image 4

(b) Image 5

(c) Image 13

(d) Image 14

(e) Image 15

Figure 1: Faces detected using the pre-trained classifier. Green - faces detected, Red - ground truth

The result of our detector applied to some of the test images can be seen in Figure 1. The True Positive Rate (TPR) is the fraction of successfully detected faces out of all valid faces. Table 1 shows the TPR for all of the test images. We can see that the detector works well but is not perfect, missing one face in Figure 1e and gives some false positives, as seen in Figure 1b.

1.1 Difficulties

It is difficult to accurately and correctly assess the TPR. One of the reasons for this is that identifying the ground truth is troublesome. Using the identification of faces as an example, what do we classify as a face? Does it include the space from eyes to mouth, or all the way from the top of the head down to the chin? What about hair? Should a face be detected if only a side profile is visible? The ambiguity of what qualifies as ground truth makes the task difficult without clear rules, but this is a different problem to our project so we will simply give a rough estimate for the face shapes.

Another difficulty comes with qualifying whether a detected face matches the ground truth. One method of measuring this is by calculating the Intersection Over Union (IOU) between them, and then comparing the result to some threshold value. This poses difficulty in deciding the value of the threshold. After some experimentation, we found that a threshold value of 0.5 meant that the most faces were detected properly without giving too many false positives.

$$IOU(A,B) = \frac{Intersection(A,B)}{Union(A,B)} \tag{1}$$

It is worth noting that it is possible to achieve a TPR of 100% by simply identifying every possible subset of an image as a face. This indicates that using the TPR to assess efficacy of a detector is not the sufficient. A measure that also takes false positives and false negatives into account is the F1 score.

$$\begin{aligned} F_1 &= 2 \cdot \frac{precision \cdot recall}{precision + recall} \\ &= \frac{2 \cdot TP}{2 \cdot TP + FN + FP} \end{aligned} \tag{2}$$

The F1 scores for this initial face detector can be seen in Table 2. As we can see, the mean result is lower because negatives are also taken into account.

2 Building and Testing A New Detector

Now we will begin building our own detector, this time identifying dartboards rather than human faces. First we use OpenCV's boosting tool to create a large set of augmented sample images of dartboards from a single prototype input. Each sample image is created by changing the viewing angle and contrast to reflect variability. Now we will use this set of positive images, along with a set of negative images stored in the `negatives` folder, to train our dartboard detector using AdaBoost.

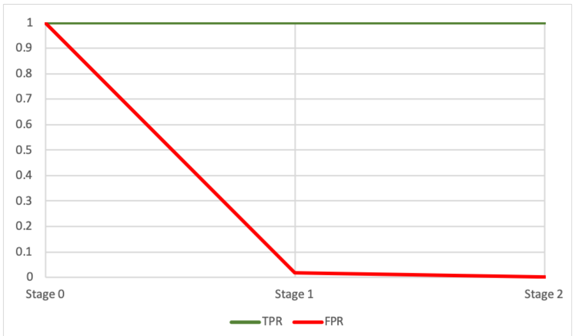
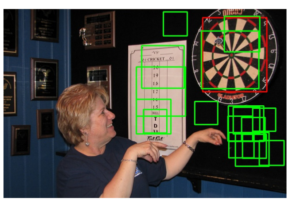


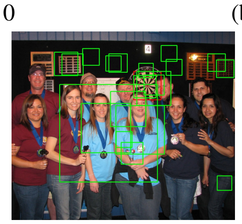
Figure 2: TPR vs FPR Achieved by the training tool

At each stage in the detector, the TPR and FPR is output, Figure 2 shows these results on a graph. At each stage, the false positive rate drops dramatically as the classifier sees more images.



(a) Image 0

(b) Image 1



(c) Image 5

Figure 3: Dartboards detected using the Viola-Jones and classifier generated

2.1 Evaluating The Detector

We can now evaluate the detector. Figure 3 shows the detectors performance on some of the test images and Table 3 shows the TPR and F1 score across all the images.

A mean TPR of 0.84 is good, but a mean F1 score of 0.17 is atrocious. Looking at Figure 3a, we can see that there are many detected dartboards which do not even resemble a dartboard to the human eye. Reasons that the TPR and FPR are much worse than in the training could be slight differences in dartboard design, obstructions and also the dramatic increase in resolution.

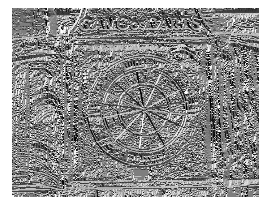
3 Integration With Shape Detectors

In order to improve our dartboard detector, we will aim to reduce the number of false positives by using shape detection. Shapes common to all dartboards are circles and intersecting lines - so we will build detectors that identify both of these.

3.1 Implementation

To detect these circles and lines, we use our magnitude (Figure 4a) and detection (Figure 4b) matrices generated by our Sobel filter. Pixels which have a value above our threshold (larger than 45/255) are considered as edges, and their corresponding hough spaces are incremented.

(a) Magnitude



(b) Direction

Both our 2D circle clustered line hough spaces are then passed to a combine function, which sums the values of each image and again normalises to values between 0 and 255. This final 2D matrix contains both the detections of circles and line clusters. Values above a threshold (larger than 240/255) signify the center of a dartboard.

3.2 Integration

Now that we have the shape detectors ready, we can integrate this with the previous dartboard detector. We decided that there are two options:

1. Run the shape detector on the whole image, and then run the Viola-Jones detector. For each Viola-Jones detected dartboard, check if the shape detector found a circle or some cluster of lines within the frame. If the shape detector meets some threshold, then accept the dartboard, else discard it.

2. Run the Viola-Jones detector first, then run the shape detector on the dartboards frame. We hypothesize that this option would be much faster, since the Hough transform only needs to be applied to a smaller area. However, this method may be less accurate since parts of circles outside the frame area would not be included.

Using method 2, we run into the issue of thresholding. Because the Viola-Jones frames are considered individually and are normalised in our hough detection algorithms, there will always be a value above 240 within the frame. This means each frame - regardless of whether it contains circles and line clusters - will be falsely detected as a dartboard. Because of this problem, we chose to opt for method 1, running the hough transform for the entire image separately from Viola-Jones before combining the two.

This is necessary for the lines to be detected properly, but means that 'weaker' lines will not be counted. Figure **??** shows an example where the lines on the dartboard are not identified. Because of this, we will opt for method 2. 2

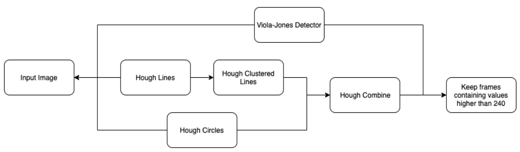
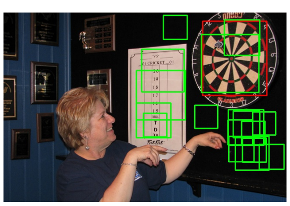


Figure 5: Flowchart showing how we integrated shape detection.

3.3 Evaluation

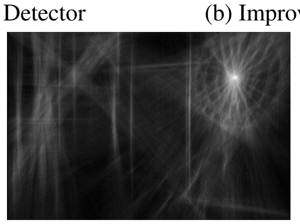
Now we are ready to evaluate the detector. Table 4 shows the TPR and F1 score when applied to the test images. The decrease in TPR can be explained by the fact that some correct frames would be discarded if the hough output is not strong enough. An example of this is is seen in Figure 8.

The F1 score has increased by a large fraction since many of the false positives have been removed, which means that by our measure, the detector is more accurate.



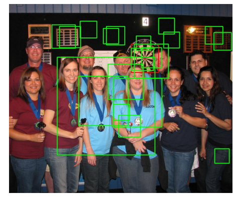
(a) Original Detector

(b) Improved Detector



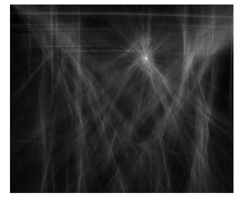
(c) Hough Space

Figure 6: Results of the original and improved detector on Image 0



(a) Original Detector

(b) Improved Detector



(c) Hough Space

Figure 7: Results of the original and improved detector on Image 5

4 Improving The Detector

Whilst our new hough detector improves upon the results obtained by the Viola-Jones method, there is still room for improvement. In this section, we will explain and implement some tweaks to further heighten the accuracy of our detector.

4.1 Image Size Reduction

We reduced the resolution of the image for our hough detections, essentially combining together pixels into chunks - this allows for lines that intersect close by to be considered as a singular point rather than a spread.

4.2 Gaussian Blur

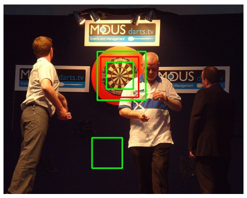
Applying a Gaussian blur filter can help to improve the accuracy of our detector as it removes noise from low frequency sections of the input image. Because both our circle and line detections have low thresholds ($> 45/255$), Gaussian blur reduces the amount of incorrect edges found by the hough transform.

4.3 Centering

When the combination of our hough transforms is normalised the highest/brightest values correspond to the center of each dartboard. We used these coordinates to re-centre our original Viola-Jones detection which meant that more dartboards were detected correctly, increasing our F1 score.

4.4 Discarding Overlapping Frames

Viola-Jones detection works by using a sliding door algorithm which considers a small part of the image at a time for detection. The algorithm is applied across the image repeatedly for various sizes of dartboard. This leads to multiple overlapping frames of the same dartboard. We used IOU to remove overlapping frames, favouring smaller sizes as they tended to be closer to ground truth. This reduced the number of false positives.



(a) Original Detector

(b) Improved Detector

Figure 8: Results of the original and improved detector on Image 6

4.5 Evaluating The Improvements

Using the methods described in sections 4.2, 4.3, **??** and 4.4, we created a detector to further improve the accuracy. Table 5 shows the results. The F1 score is only slightly higher. We think that the reason for this is because when we remove overlapping frames and pick the smallest one, they are too small. To improve, we could select the median size.

5 Future Improvements

5.1 Improving How We Test

Using the F1 score assumes that the recall is equally as important as precision. An F_X measure is used where recall is X times more important than precision. We could have thought more about how we would weigh this in a real situation. If we repeated the project we would have also greatly increase the size of the test set, in order to obtain a more accurate score.

5.2 Deep Neural Network

A different improvement to this system involved training a deep neural network with the Keras[2] framework, using a TensorFlow[3] back-end, on a manually created dataset of 200 dartboard images 200 random images. The network is sequential, and contains alternating Conv2D / MaxPooling2D layers with 'relu' activation functions, connected to a final dense 'sigmoid' layer which outputs the probability of a dartboard being present in a given image. As the network is trained, the weights of each layer are adjusted through the back-propagation algorithm to favour gradient descent. This weight adjustment can be thought of as the network learning local patterns interpretations within the dartboard samples (i.e. circles lines). After training for 10 epochs, our model was validated using the original 16 dartboard images, resulting in a validation accuracy of 0.97 and a validation loss of 0.25. The results shown in Figure 9 show that a maximum accuracy is achieved around the 9th epoch. Due to a small validation set, the validation plot is noisy and therefore should not be taken as the true accuracy of the model.

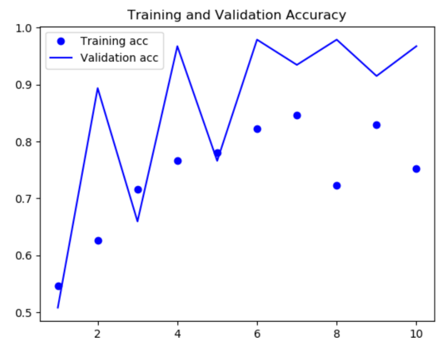


Figure 9: Training and validation accuracy for the network

Unfortunately, we struggled to convert our Keras model (which runs on Python) into C++, so were unable to use it in our detector. Given more time, this network would ideally then be used in conjunction with our hough transforms to improve the accuracy of the Viola-Jones detection.

5.3 Ellipses

An ellipsis hough transform would have aided the detection of slanted dartboards in images such as dart10.jpg. The hough space would contain 5 dimensions (X, Y, Radius, Rotation, Squish).

6 Appendix

Image	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Mean
TPR	0	0	0	0	1	1	0	1	0	1	0	1	0	1	1	0.33	0.46

Table 1: TPR of face detector over all test images (2 d.p)

Image	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Mean
F1	0	0	0	0	1	0.88	0	1	0	0.4	0	1	0	0.67	0.44	0.29	0.35

Table 2: F1 score of face detector over all test images

Image	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Mean
TPR	1	1	1	1	1	1	1	1	0.5	1	0.67	1	0	1	0.5	1	0.84
F1	0.11	0.67	0.22	0.2	0.14	0.10	0.14	0.08	0.05	0.22	0.12	0.4	0	0.2	0.03	0.25	0.17

Table 3: TPR and F1 score of initial dartboard detector over all test images

Image	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Mean	Difference
TPR	1	1	1	1	1	0	0	1	0.5	1	0	1	1	0	0	0	0.59	-0.26
F1	1	1	0.67	1	0.67	0	0	0.67	0.4	0.67	0	1	0.5	0	0	0	0.47	+0.29

Table 4: TPR and F1 score of the improved dartboard detector over all test images, and the difference to the original detector

Image	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Mean	Difference
TPR	1	1	1	1	1	1	0	0	0.5	1	0	0	0	0	0.5	0	0.5	-0.09
F1	1	1	1	1	1	1	0	0	0.67	1	0	0	0	0	0.5	0	0.51	+0.04

Table 5: TPR and F1 score of the further improved dartboard detector over all test images, and the difference to the previous detector

Zack	1.0
Jake	1.0

Table 6: Contribution weights

References

[1] P. Viola and M. Jones, "Robust real-time object detection," in *International Journal of Computer Vision*, 2001.

[2] F. Chollet *et al.*, "Keras." `https://keras.io`, 2015.

[3] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.