The Darts Challenge

Louis Auger, Maria Marinova

Object detection and recognition is an important task in Computer Vision. This paper demonstrates two such algorithms — Viola Jones and Hough Transform.

1. Viola-Jones Object Detector

Figures 1–5 demonstrate the Viola-Jones Object Detector's ability to detect faces using Haar-like features. The detector seems to have difficulty with viewpoint changes (Fig. 5). It is also worth noting that some objects are incorrectly classified as faces as they possess a number of Haar-like features, for example the two darts in the lady's hand in figure 2 are incorrectly considered as eyes.

Figure 1: *dart4.jpg*

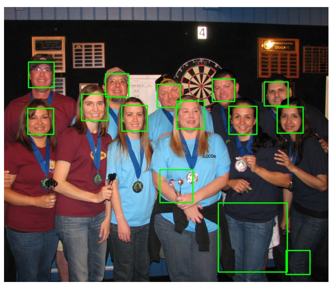

Figure 2: *dart5.jpg*

Figure 3: *dart13.jpg*

Figure 4: *dart14.jpg*

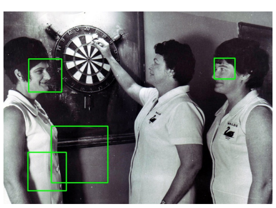

Figure 5: *dart15.jpg*

Figures 6–9 demonstrate the Viola-Jones Object Detector's limitations on a test set that we made. We target performance with regards to occlusion, in-plane rotation, differing lighting and viewpoint changes (in this order).

One way to calculate the accuracy of a classifier — in this case the Viola-Jones Object Detector — is calculating the true positive rate (TPR). In table 1 we show the TPR for images *dart5.jpg* (Fig. 2) and *dart15.jpg* (Fig. 5) using the formula $TPR = \frac{\text{true positives}}{\text{all positives}}$.

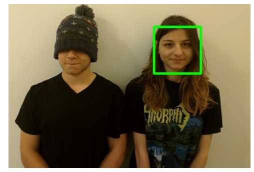

Figure 6: *Partial Occlusion*

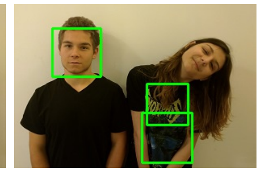

Figure 7: *In-plane Rotation*

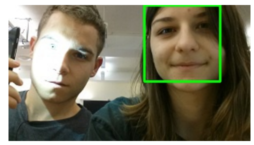

Figure 8: *Differing Lighting*

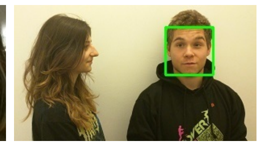

Figure 9: *Viewpoint Changes*

Table 1: *True Positive Rate of dart5.jpg and dart15.jpg*

Image	TP	P	TPR	%
dart5.jpg	11	11	1	100%
dart15.jpg	2	3	0.67	67%

As demonstrated, the true positive rate is not a sufficient performance indicator: the TPR of *dart5.jpg* is 100% even though there are 3 *false positives*, i.e. 3 areas of the image are incorrectly categorized as faces. Moreover, the detector sometimes detects only half a face, giving an ambiguous classification.

Additionally, a TPR of 100% is trivially achieved on any detection task by classifying everything as positive.

It is apparent that the TPR does not capture performance on the *false positives* and *true negatives*. One may calculate the *false positive rate*. However, when working with images, calculating the *true negatives* is computationally infeasible, as there are numerous combinations of pixels to consider. Instead, one may calculate the *false positives*, and from this, calculate the *precision*.

$$precision = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

Table 2: *Precision of dart5.jpg and dart15.jpg*

Image	TP	FP	precision
dart5.jpg	11	3	$\frac{11}{11+3} = 0.79$
dart15.jpg	2	2	$\frac{2}{2+2} = 0.5$

2. Building and Testing a Detector

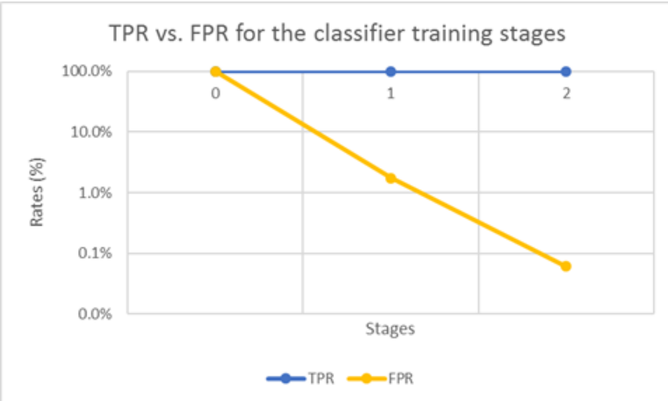

Figure 10: *TPR vx. FPR*

Figure 10 shows a graph of the achieved TPR and FPR at each stage of building the strong classifier. New features are added at each stage, exponentially decreasing the likelihood that an instance will exhibit all features, hence reducing the FPR in the way seen.

The TPR is 100% at each stage, so we can expect any dartboard that closely resembles the positive training image to be positively classified. Whilst the FPR is not 100% (it is not high, and reduces with each stage), a small number of image areas are expected to be misclassified as dartboards, to avoid overfitting.

Figures 11–18 show the performance of the classifier on eight of the images in the provided set.

The classifier performance on this test data followed the expectations from the graph in figure 10:

- High TPR when dartboards resemble a training data dartboard image (Fig. 11-13)
- Low TPR when the dartboards are subject to:
  - Differing lighting (Fig. 14)
  - Partial occlusion (Fig. 15)
  - Viewpoint changes (Fig. 17)
- There are some false positives: the main cause being normalising lighting (Fig. 16, 18)

The following conclusions about predicting system performance on unknown test data may be extrapolated from figure 10:

- Whilst more training stages would drop the FPR at a logarithmic rate, they would quickly give diminishing returns as is characteristic of classifiers based on boosting.
- As a consequence of the attentional cascading, more stages may also decrease the TPR as rejection will gradually be more likely
- As with all training procedures, excessive training leads to overfitting

Figure 11: *dart1.jpg*

Figure 12: *dart3.jpg*

Figure 13: *dart4.jpg*

Figure 14: *dart6.jpg*

Figure 15: *dart7.jpg*

Figure 16: *dart9.jpg*

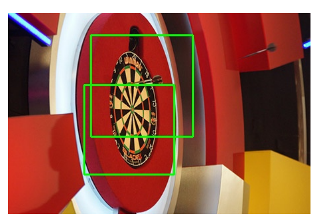

Figure 17: *dart12.jpg*

Figure 18: *dart14.jpg*

3. Integration with Hough Transforms

Table 3: *Dartboard Detector Performance*

Image	TPR	FP	Comments
dart4.jpg	1/1	1	FP on striped shirt
dart8.jpg	1/2	0	one dartboard missed
dart10.jpg	4/3	0	one dartboard detected twice
dart11.jpg	0/1	0	one dartboard missed
the rest	100%	0	ideal performance

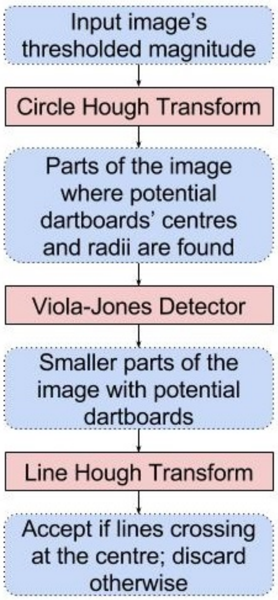

Figure 19: *Initial idea*

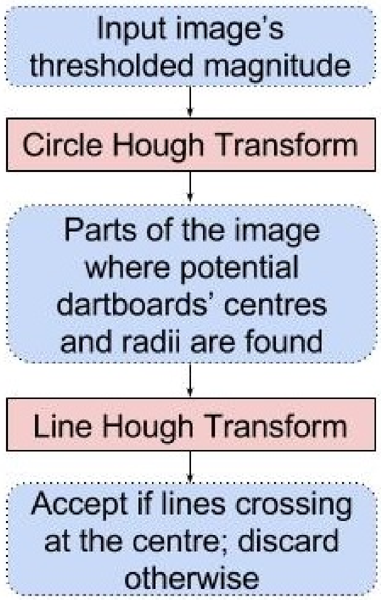

Figure 20: *Final implementation*

Key merits and limitations of the dartboard detector:

+ Partially occluded dartboards are detected (Fig. 27)
+ Dartboards subject to viewpoint changes are detected
– Sensitive to differing lighting (Fig. 28)
– Sensitive to noise

We implemented both the Circle and Line Hough Transform, and experimented with the Viola-Jones Detector, before deciding on a way to combine them for dartboard detection. Our final implementation is a result of a number of observations and performance testing:

* We do not start with Viola-Jones as it did not detect the dartboards missed by the Circle Hough Transform, yet rejected 50% of the TP detected by the Circle Hough Transform
* Circle Hough Transform captured all TP (except for dart8, dart11) and a number of FP and was therefore chosen to locate possible dartboards
* The Line Hough Space was difficult to threshold unless the dartboard covers the majority of the image, as longer lines are stronger, and was therefore used to verify the areas found by the Circle Hough Transform

Thresholded Gradient Magnitude

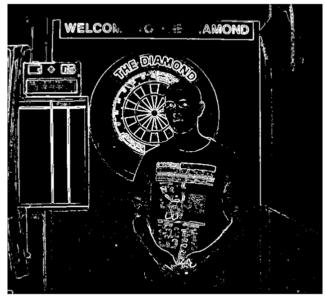

Figure 21: *dart7*

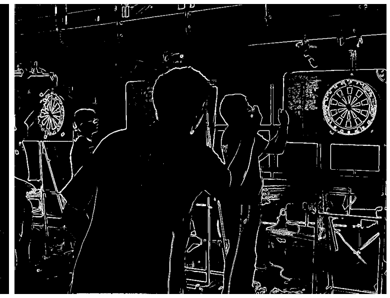

Figure 22: *dart8*

Circle Hough Space (x, y, r)

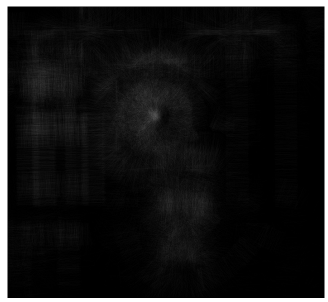

Figure 23: *dart7*

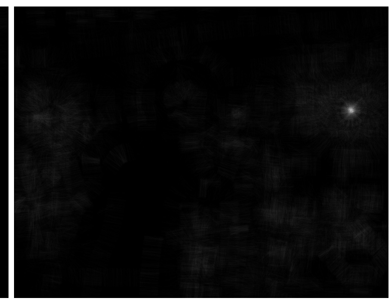

Figure 24: *dart8*

Line Hough Space (ρ, θ)

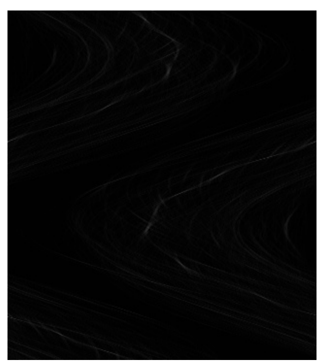

Figure 25: *dart7*

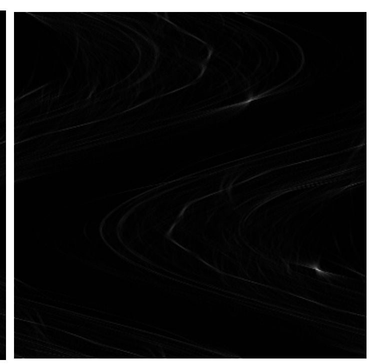

Figure 26: *dart8*

Detected Dartboards

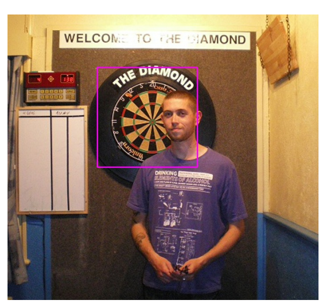

Figure 27: *dart7*

Figure 28: *dart8*

4. Building and Testing a Detector

Harris corner detection

There are several reasons that we chose to add corner detection to our dartboard detector:

* We needed a feature to reduce the false positives
* Many edges of a dartboard are connected at near perpendicular angles, forming corners
* Near-parallel lines close to each other create many 'psuedo' intersections, potentially leading to false positives

Figure 29: *dart4*

Figure 30: *dart9*

Table 4: *Improved Dartboard Detector Performance*

Image	TPR	FP	Comments
dart8.jpg	1/2	0	one dartboard missed
dart10.jpg	4/3	0	one dartboard detected twice
dart11.jpg	0/1	0	one dartboard missed
the rest	100%	0	ideal performance

The following key merits and shortcomings were witnessed:

+ Reduced the number of false positives (where there are circles and lines, but no corners)
+ True positive rate not reduced
− Detector is still dependent on the performance of Circle Hough Transform

Other improvement ideas

Ellipse/Half-circle detection

As dartboards are initially located based on the Circle Hough Transform, an ability to detect half-circles and ellipses would improve the locating of dartboards subject to viewpoint changes and occlusion. While this implies a larger amount of FP being suggested at this interim step, a local Line Hough Transform would be able to discard the (majority of) FP.

Sliding window

As observed, detecting lines for dartboard detection is only sufficient if there are not stronger lines present in the image — which is rarely the case for a full image. Therefore, a method for dartboard detection is to check for crossing lines in windows of size 100x100, overlapping in the middle, and looking for circles only when crossing lines are detected. This will, however, mean that the algorithm will be run many times for a single image, and would therefore be slow.

Colour histograms

As dartboards generally follow the same colour scheme, histograms may be used for colour evaluation in order to determine whether an object is a dartboard, based on colour detection of a trained image. It is important to note that this would only work on a local area that consists (almost) exclusively of the dartboard, as otherwise the colours in the rest of the image would introduce far too much noise.

Conclusion

In conclusion, there are numerous way to combine simple detection algorithms in order to detect complicated objects. Our work demonstrates one such combination to detect dartboards with a high success rate.