COMPUTER VISION COURSEWORK 2023

George Atkinson

Part I – Shape Detection

In this section, I will detail how I used object detection methods in order to create a system for detecting dartboards in images.

Subtask 1 – Viola-Jones Detector

Viola-Jones detection is a common method in computer vision that looks for Haar-like features in objects for detection in images.

Each time the classifier iterates over the training data it uses the AdaBoost algorithm to look for better features for detecting dartboards.

To train a Viola-Jones detector I used a single prototype image as input to opencv_createsamples to create 500 more positive samples with added noise and distortion. Then, I used opencv_traincascade with an input of positive and negative samples to train the Viola-Jones classifier to detect dartboards.

Training Performance

Upon each iteration of the training data, the model outputs data on its performance. This data includes the true positive rate (TPR), false positive rate (FPR), and the training time. These metrics serve as a great way of analyzing the performance of the Viola-Jones cascade. To begin with, I analyzed how the FPR was affected by each training stage.

The results of this can be seen in figure 1 in which I have plotted the number of stages against the natural logarithm of the FPR and TPR.
From this, I could see a clear negative correlation between training stages and FPR – i.e., the more stages, the more accurate the model performs. In addition, it's clear to see from the figure that with each stage the TPR remains the same – at a value of 1 (or 0 on the logarithmic scale used in the graph).

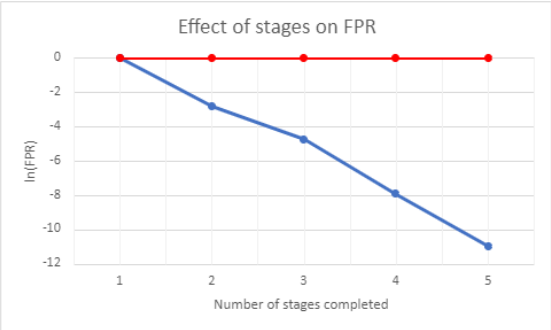

Figure 1: a graph showing the relationship between training stages, FPR and TPR

However, this improvement in accuracy comes at a cost. As a result of performing more iterations on the training data, it unsurprisingly took significantly longer to train the cascade. This effect can be seen in figure 2 in which I plot the number of stages against the runtime of opencv_traincascade.

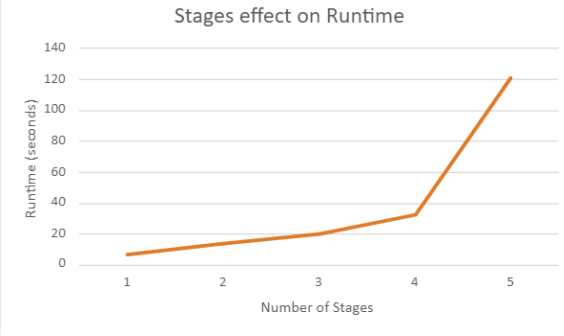

Figure 2: a graph showing the relationship between training stages and the runtime of opencv_traincascade.

From the graph in figure 2, it is clear to see that the training time takes a significant increase to get more accurate results. In fact, it takes approximately 8 times longer to run the cascade for 5 stages instead of 2. For this reason, I think it is best to run it for 3 stages as this achieves a very good FPR whilst keeping the runtime low.

Testing performance

Now to evaluate the performance of my Viola-Jones classifier – I got my testing dataset containing 16 images containing at least 1 dartboard, and then collected ground truth data for the location of the dartboards in each image.

Figure 3: test images dart0.jpg, dart4.jpg, and dart9.jpg (bottom left, top left, right) with ground truth (red) and Viola-Jones (green) bounding boxes.

Using the ground truth data as well as the Viola-Jones detections, I drew bounding boxes for the dartboards with ground truth in red and Viola-Jones in green. Some examples of this can be found in figure 3 and by a brief look you can see that whilst the Viola-Jones model consistently detects dartboards, it also detects a lot of false positives.

To decide if a detected dartboard was correct, I classified it as a true positive if its IOU with the ground truth is greater than 0.5.

With this, I was then able to evaluate the performance of my Viola-Jones model by calculating the TPR and the F1-score for each image. By calculating these scores and averaging them across the test set I found that this model averaged a TPR of 0.43 and an F1-score of 0.08.

Image	TPR	F1-score
Dart0.jpg	1.00	0.13
Dart1.jpg	0.00	0.00
Dart2.jpg	1.00	0.14
Dart3.jpg	1.00	0.22
Dart4.jpg	0.00	0.00
Dart5.jpg	1.00	0.11
Dart6.jpg	0.00	0.00
Dart7.jpg	0.00	0.00
Dart8.jpg	0.50	0.16
Dart9.jpg	0.00	0.00
Dart10.jpg	0.00	0.00
Dart11.jpg	0.33	0.29
Dart12.jpg	0.00	0.00
Dart13.jpg	1.00	0.13
Dart14.jpg	1.00	0.06
Dart15.jpg	0.00	0.00
Average	0.43	0.08

Figure 4: a table showing the TPR and F1-score for the cascade on each test image.

Whilst some of the TPR scores in figure 4 are quite good, there is a significant number of poor scores. The reason for the disparity in TPR between the testing and training data is because of how I defined the ground truth, as well as how I identified true positives. When defining ground truth, I defined a dartboard to include the black border of the board whereas the cascade would only look for the inner target part of the dartboard. This meant that when I classify my true positives, even if the cascade has detected the target, it may not be labelled as a true positive since there is an IOU of less than 0.5. An example of this can be seen with Dart9.jpg where in figure 3 you can see a strong bounding box of the dartboard, but, in figure 4 it has a TPR of 0.

Furthermore, the F1-scores are cohesively bad amongst all the test images. This is mostly because of the extremely large number of false positives in the images, but it is also due to the aforementioned issue with calculating the true positive rate.

Subtask 2 – Integration with Shape Detectors

Now with a base model for detecting dartboards, I will attempt to improve the performance by using Hough transformations to detect distinct features in the dartboards. Then, with this I will combine my results with the Viola-Jones model to hopefully improve the TPRs and F1-scores.

Hough Details

Then with a base model for detecting dartboards, I attempted to improve the model using Hough transformations to detect distinguishing patterns in the dartboard.

After preprocessing and Sobel filtering each image I used Hough line detection with support from my Viola-Jones detector to detect lines in the image. I excluded any horizontal or vertical lines from my Hough space because they tended to be other prominent features of the images e.g., walls, doors, etc.

Then with, some help from the Viola-Jones detections, I found points of intersections between lines that met at an angle of around 18 degrees. This was an idea to help filter lines since all the dartboards (that had no parallax error) would have their target lines meet at this angle.

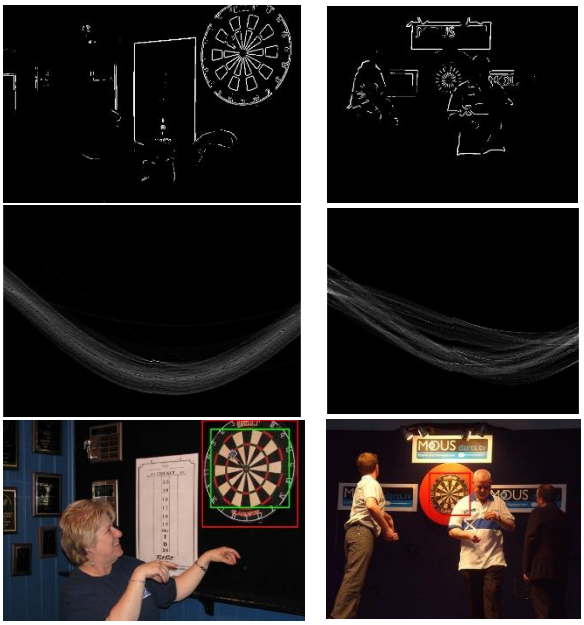

Figure 5: Thresholded images, 2D Hough space representations, and output images for Dart0.jpg (left) and Dart6.jpg (right)

One of my best outputs can be seen in Figure 5 (left side) where a clear Hough line can be seen, and it results in a very strong detection. However, my model was not good with other images which had other prominent lines in the image – especially when the dartboard was small and noisy.

Image	TPR	F1-score
Dart0.jpg	1.00	1.00
Dart1.jpg	0.00	0.00
Dart2.jpg	1.00	1.00
Dart3.jpg	0.00	0.00
Dart4.jpg	1.00	1.00
Dart5.jpg	1.00	1.00
Dart6.jpg	0.00	0.00
Dart7.jpg	1.00	1.00
Dart8.jpg	0.00	0.00
Dart9.jpg	1.00	1.00
Dart10.jpg	0.00	0.00
Dart11.jpg	0.33	0.60
Dart12.jpg	0.00	0.00
Dart13.jpg	1.00	1.00
Dart14.jpg	0.50	0.50
Dart15.jpg	1.00	1.00
Average	0.55	0.57

Figure 6: a table showing the TPRs and F1-scores after combining my Viola-Jones boxes with Hough transformations.

Evaluation

After reviewing my results from figure 6 and comparing them with the previous results from figure 4, I can conclude that my new model has improved the TPR and especially the F1-scores. There is still a notable amount of 0-valued TPR scores, but this is again due to the ground truth boxes being too large such that an IOU of 0.5 rules out a detection that generally looks good. An example of this is in figure 7 where the detection looks good, but it is unfortunately not considered a true positive.

Figure 7: a successful detection harshly labelled as a false positive.

Merits of system:

- Does excellent job at removing false positives.
- Good at finding centres of dartboards, even in cluttered images.
- Moderate success at detecting obstructed dartboards.

Drawbacks of system:

- Not as good at detecting smaller dartboards or dartboards at more of an angle.
- No way of gauging the size of the dartboard for adjustment of bounding box – implementing this would greatly increase the TPR.
- When Viola-Jones doesn't find a dartboard, the line detector can't either.

Detection Pipeline

To combine my Viola-Jones model and my Hough transformations – my aim was to remove as many false positives as possible whilst keeping true positives.

When I searched for intersections, I only looked for intersections that were within a Viola-Jones boundary box. This improved the performance of my model significantly because it meant that lines from opposite sides of the images would interact less.

Also, I then found the largest Viola-Jones box near to my detected center and then repositioned it to be centered around my new center. T

In addition, I enlarged my detection boxes by 20% to account for the issue I was having with my ground truth being larger than the Viola-Jones box detections.

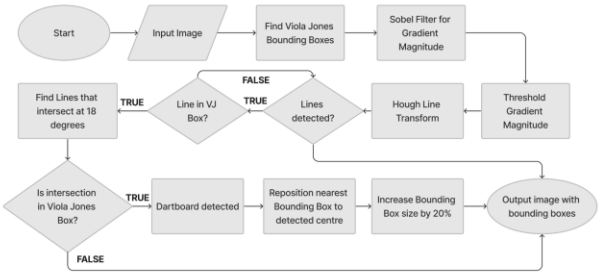

Figure 8: Detection Pipeline outlining how images are processed with the help of Viola-Jones detections.

Potential Improvements

Despite the success of my system, it is natural to think of improvements that could have been made with more time. The next steps I would have taken to improve my code are:

- Remove any gradient magnitude outside of Viola-Jones box to make line detection easier.
- Implement another object detection system such as YOLO to help filter false positives and make system less dependent on Viola-Jones.