Image Processing and Computer Vision
The Dartboard Challenge

Student id: jp17528

Introduction

Viola Jones algorithm is one of the most renowned image processing tools for detecting faces. The algorithm can detect faces in a random image instantly by using four main techniques: Haar feature selection, integral image, adaboost algorithm, and cascading classifiers.

Firstly, this paper will use a given classifier, *frontalface.xml* generated by Viola Jones algorithm for detecting faces and evaluate its effectiveness. Secondly, the paper will implement a generated classifier *cascade.xml* by the algorithm for detecting given dart board images and evaluate its performance. Thirdly, the paper will introduce a method to detect dartboards not only using Viola Jones algorithm but also using two types of Hough transform: line detection Hough transform and circle detection Hough transform. Lastly, the paper will introduce a possible way to further improve the detection as well as providing possible recommendations for better detection.

Subtask 1: Viola-Jones Object Detector

The program detected faces with given classifier, frontalface.xml. These are the following results for dart4.jpg, dart5.jpg, dart13.jpg, dart14.jpg, and dart15.jpg.

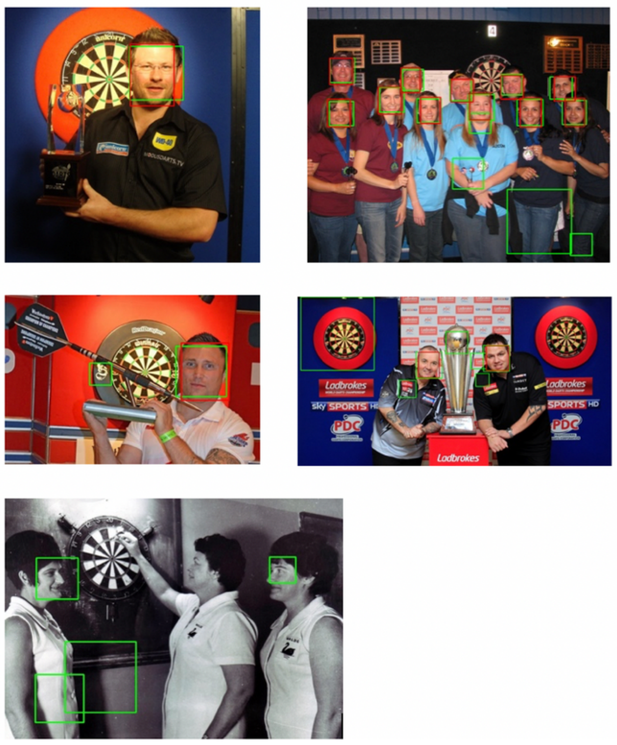


Figure 1. Five example images of frontal face detection.

This following table shows TPR and F1 score of each image.

	TPR	F1 score	Frontal face
dart0.jpg	0	0	No
dart1.jpg	0	0	No
dart2.jpg	0	0	No
dart3.jpg	0	0	No
dart4.jpg	**1**	**1**	**Yes**
dart5.jpg	**1**	**0.846**	**Yes**
dart6.jpg	**0**	**0**	**Yes**
dart7.jpg	**1**	**1**	**Yes**
dart8.jpg	0	0	No
dart9.jpg	**1**	**0.4**	**Yes**
dart10.jpg	0	0	No
dart11.jpg	**1**	**1**	**Yes**
dart12.jpg	0	0	No
dart13.jpg	**1**	**0.667**	**Yes**
dart14.jpg	**1**	**0.444**	**Yes**
dart15.jpg	0	0	No

To judge whether the frontal face is correctly detected or not, intersection-over-union (IOU) is used. IOU takes intersection of detected faces (green) and ground truth faces (red) over union of detected faces (green) and ground truth faces (red). A threshold of IOU is selected as 0.5 indicating that it will classify as face when IOU area is over than 50%.

The average value of the TPR and the F1 score for the images which has the frontal face is 0.875 and 0.670. At first glance by analysing the TPR, face detection seems to work well. However, it is difficult to assess the TPR meaningfully. Firstly, it is always possible to get TPR as 1 by changing various parameters in *detectMultiscale* function such as *scaleFactor*, *minNeighbors*, *minSize*, and *maxSize*. It is worth noting that TPR of dart6.jpg is 0. This is due to the minSize parameter being too large to detect the frontal face in dart6.jpg. It was possible to get TPR = 1 when *minSize* is changed to (25, 25). Secondly, the ground truth red square and IOU threshold are subjective. Therefore, it is more reasonable to focus on F1 score since this score takes precision in to account. F1 score is lower than TPR because precision considers false positives. A method to reduce false positives will be covered in detail in section 3 by using dartboards as an example.

TPR = TP / (TP + FN)

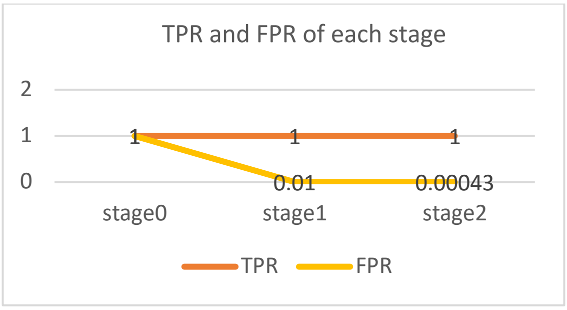
precision = TP / (TP + FP)

TP: True Positive, FN: False Negative, FP: False Positive

Subtask 2: Building & Testing your own Detector

2-1. Training performance

To utilize Viola Jones framework as mentioned in the introduction, the dartboards image set (positive set) and non-dartboards image set (negative set) have been generated. In the positive set, various shape of dartboards was created; thus, increasing the robustness of detection. The graph below illustrates how TPR and FPR change in each stage of cascades.



TPR remains 1 for all stages which indicates that the training algorithm detects every dartboard. FPR decreases at each stage discarding false positives using attentional cascade; thus, rendering better classifier.

2-2. Testing performance

First of all, the parameters *scaleFactor*, *minNeighbors* of *detectMultiSciale* are changed because the detection object has been changed from frontal face to dartboards. The new parameters were selected to meet the following criteria: 1. Try to find at least one dartboard in the image. 2. Do not generate too many false positives since this will lose capability of the dartboard classifier itself.

	TPR	F1 Score
dart0.jpg	1	0.286
dart1.jpg	1	0.666
dart2.jpg	1	0.333
dart3.jpg	1	0.4
dart4.jpg	1	0.4
dart5.jpg	1	0.375
dart6.jpg	1	0.5
dart7.jpg	1	0.25
dart8.jpg	0.5	0.167
dart9.jpg	1	0.182
dart10.jpg	0.666	0.333
dart11.jpg	1	0.666
dart12.jpg	1	0.666
dart13.jpg	0	0
dart14.jpg	0.5	0.105
dart15.jpg	1	0.666
Average	0.854	0.375

By using the criteria mentioned earlier, it is not possible to get average TPR=1 for testing images. This is because the cascade has been implemented to the new testing data. In fact, it was possible to get TPR=1 for all the images; however, this will increase false positives violating the second rule of criteria.

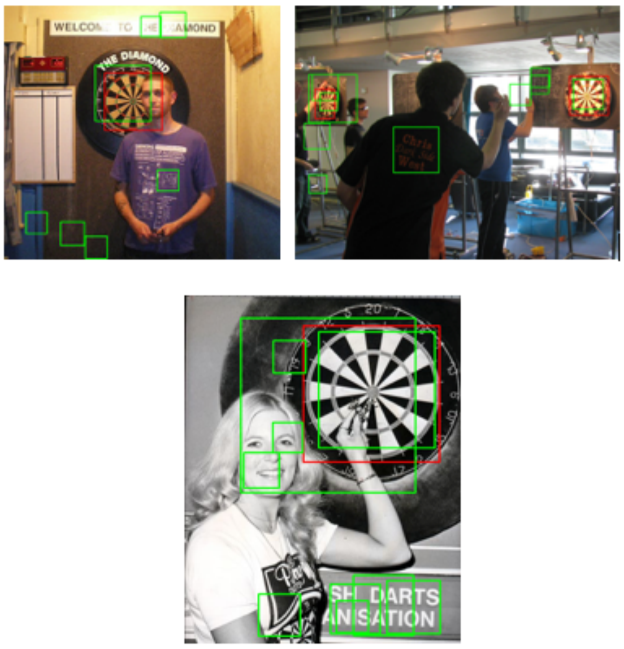


Figure 2. dart7.jpg, dart8.jpg, and dart9.jpg

It turned out that dartboard detection cascade model ended in failure by analyzing F1 score (IOU threshold=0.5) and *figure2*. The model generated many false positives. There are a few reasons why the model generated many false positives.

- *opencv_createsamples* did not generate enough training examples as there were only 500 positive dartboards examples.
- *opencv_createsamples* did not generate good enough haar like features to distinguish between dartboards and non-dartboards.
- *opencv_traincascade* did not use enough negative sample images as number of negative images (*numNeg*) was 500.
- Testing images are noisy. Lighting and angle vary depending on the images.

In the next section, this paper will introduce how to reduce false positives by using Hough transform leading to increase F1 score closer to 1.

Subtask 3: Integration with Shape Detectors

3-1. Detection pipeline

To decrease false positives, the program implemented both Hough line transform and Hough circle transform.

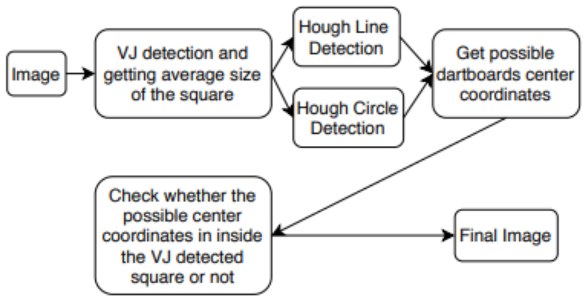


Image
VJ detection and getting average size of the square
Hough Line Detection
Hough Circle Detection
Get possible dartboards center coordinates
Check whether the possible center coordinates in inside the VJ detected square or not
Final Image

Figure 3.2 Flow diagram of the procedures to detect dartboards.

- The program reads an image and generates the thresholded gradient magnitude image.
- Run Viola Jones detection algorithm to detect dartboards and get average size of the square. This will be used later for Hough circle transform. By acquiring the average, it is possible to get the appropriate radius range per image. This will significantly increase the performance of the program since the time complexity of the circle transform is $O(N^3)$.
- The program uses the Hough transforms to detect possible centers of the dartboards by using the edge detected image. Hough line assumes a pixel as a center when more than 3 lines are crossing the pixel (+ the line is drawn for top 10% bright pixel in Hough line space). Hough circle assumes a pixel as the center if the pixel is top 20% bright pixel in Hough circle space. By using the average above, it searches for 80 radii per image.
- If one of the candidates (generated by using Hough line or Hough circle) are located within the Viola Jones (VJ) detected square, then the program detects that square as a dartboard.

3-2. Hough Details

By observing figure4 and figure5, it is possible to conclude that both Hough line and circle transform worked well to detect the dartboard. In dart8.jpg(figure4), F1 score increased from 0.167 to 0.5. Likewise, in dart9.jpg(figure5), F1 score increased from 0.182 to 0.5, reducing the false positives. (Check figure2 for removed false positives) However, Hough line did not work well on dart8.jpg. Moreover, Hough circle and line detection did not work well in some images. Hough circle did not work well because the dartboards were partly shown or slanted. Hough line did not work well because firstly, there were many straight lines going on the image. Secondly, the threshold to draw a line was very high (top 10%)

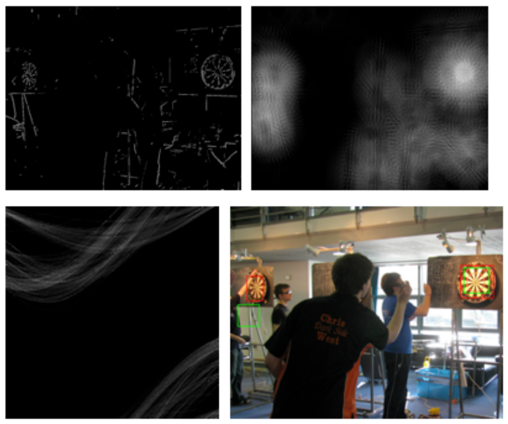


Figure 4. dart8.jpg

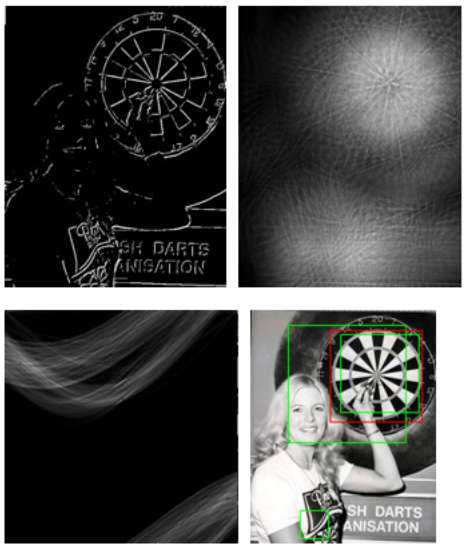


Figure 5. dart9.jpg

3-3. Evaluation

	TPR	F1	VJ	VJ+HT
dart0.jpg	1	0.666	6	2
dart1.jpg	1	0.666	2	2
dart2.jpg	1	1	5	1
dart3.jpg	1	0.4	4	4
dart4.jpg	1	1	4	1
dart5.jpg	1	0.5	13	9
dart6.jpg	0	0	3	0
dart7.jpg	1	1	7	1
dart8.jpg	0.5	0.5	10	2
dart9.jpg	1	0.5	10	3
dart10.jpg	0.333	0.5	9	1
dart11.jpg	1	1	2	1
dart12.jpg	1	0.666	2	2
dart13.jpg	0	0	3	3
dart14.jpg	0.5	0.1052	17	17
dart15.jpg	1	0.666	2	2
Average/Total	0.771	0.573	99	51

- Average F1 score increased from 0.375 to 0.573
- Detected dartboards decreased from 99 to 51
- Average TPR decreased from 0.854 to 0.771.

These facts imply that VJ with HT worked well. TPR decreased because HT did not work well for some images. It would be possible to get TPR=0.854 like subtask2 but considering F1 score, the overall quality of detection has been greatly increased.

Subtask 4: Improving your Detector

By using both VJ framework and Hough transforms, it is possible to get a better F1 score. In fact, many squares classified as false-positives are located around the dartboard. There are many possible approaches including these ideas below to improve the performance of the detector.

4-1. Possible ideas for better detection

• Try all radius when using Hough circle.

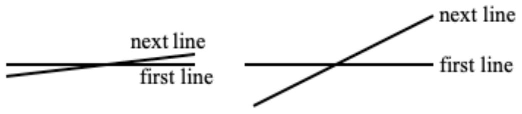
When creating the program, *speed-accuracy trade-off* has always been considered. The program did not search all the possible radius because it would significantly increase the computation time. By trying all the radius, it would be possible to detect dartboard's center better.

• Use *AND* instead of *OR* for final detection

The program detects the square as a dartboard when either center computed by Hough circle or center computed by Hough line is inside the VJ square. Instead of this, it is possible to make the program detect the square as a dartboard when both of Hough circle and Hough line detected line are inside of the VJ square.

• Make changes to Hough line detection.

Hough lines did not work well for the subtask3 because there were many other straight lines. This can be solved by drawing the first line and then drawing the line over a certain angle when drawing the next line. (LHS: current implementation, RHS: possible implementation)



• YOLOv3 object detection

Instead of using VJ framework, it is possible to use YOLO (You Only Look Once) object detection. There are many object detection models: such as SSD and R-CNN. However, YOLO was chosen because *detection speed* became one of the most important factors as well as *accuracy* since new technologies evolved (e.g. self-driving car). This task was to detect dartboards in a static image; therefore, a subtask4 program is developed to detect dartboards only for the static image. It is worth noting that YOLO dartboard detector would work in the video as well. In addition, it is possible to detect multiple objects (e.g. person and dartboard).

4-2. YOLO dartboard detection

4-2-1. How YOLO object detection works

1. Read an image and divide the image by N X N cells
2. Each cell is accountable for B bounding boxes. (Bounding box contains an object)
3. YOLO outputs a confidence score, how confidence the box contains the object. The score does not have information about what object it is.
4. The cell predicts a class of objects (here we have only one object: dartboard) for each bounding box.
5. Combine the prediction from cells (3) and confidence score (4).
6. Leave the bounding box that the confidence is over than 30% and discard rest of the bounding boxes.

4-2-2. Evaluation and Visualization (IOU=0.5)

	TPR	F1	VJ+HT	YOLO
dart0.jpg	1	1	2	1
dart1.jpg	1	1	2	1
dart2.jpg	1	1	1	1
dart3.jpg	1	1	4	1
dart4.jpg	0	0	1	0
dart5.jpg	1	1	9	1
dart6.jpg	0	0	0	0
dart7.jpg	1	1	1	1
dart8.jpg	0.5	0.5	2	1
dart9.jpg	1	1	3	1
dart10.jpg	0	0	1	0
dart11.jpg	0	0	1	0
dart12.jpg	1	1	2	1
dart13.jpg	1	1	3	1
dart14.jpg	1	1	17	2
dart15.jpg	1	1	2	1
Average/Total	0.719	0.719	51	13

• The YOLO detection speed (~1sec) was exceptionally better than VJ+HT detection method (2~3mins).
• F1 score increased from 0.573 to 0.719
• Detected dartboards reduced from 51 to 13

YOLO worked very well, detected dartboards were not counted because manual box coordinates and IOU threshold are subjective. The detector has been trained with 239 images. By increasing the training set, adjusting the IOU threshold, and the coordinates, it would be possible to increase F1 score.

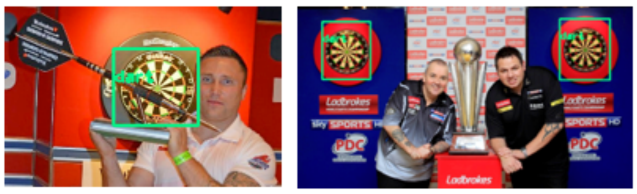


Figure 6. dart13.jpg and dart14.jpg by YOLO dartboard detection

References (report + code):

[1] Train YOLO to detect a custom object, URL: https://pysource.com/2020/04/02/train-yolo-to-detect-a-custom-object-online-with-free-gpu/

[2] YOLO object detection, URL: https://www.youtube.com/watch?v=4eIBisqx9_g

[3] Circle Hough Transform, URL: https://en.wikipedia.org/wiki/Circle_Hough_Transform

[4] How Circle Hough Transform works: URL: https://www.youtube.com/watch?v=Ltqt24SQQoI

[5] How to implement circle detection by using circle equation? URL: https://stackoverflow.com/questions/43884385/how-to-implement-circle-detection-by-using-circle-equation

[6] Hough Line Transform, URL: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_houghlines/py_houghlines.html

[7] Hough transform, URL: https://en.wikipedia.org/wiki/Hough_transform

[8] How Hough Transform works, URL: https://www.youtube.com/watch?v=4zHbI-fFIlI

[9] Hough Line Transform, URL: https://docs.opencv.org/3.4/d9/db0/tutorial_hough_lines.html

[10] Intersection over Union for object detection, URL: https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

[11] Viola–Jones object detection framework, URL: https://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework

[12] Robust Real-time Object Detection, URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.110.4868