# COMS30121 - Image Processing and Computer Vision (2016-17)
# The Object Detection Challenge

## Subtask 1: The Viola-Jones Object Detector

The first task is to use a Viola-Jones detector had been trained to recognize human faces and test it on a subset of images containing human faces. As follows, Figure 1 is the results on five given example images.

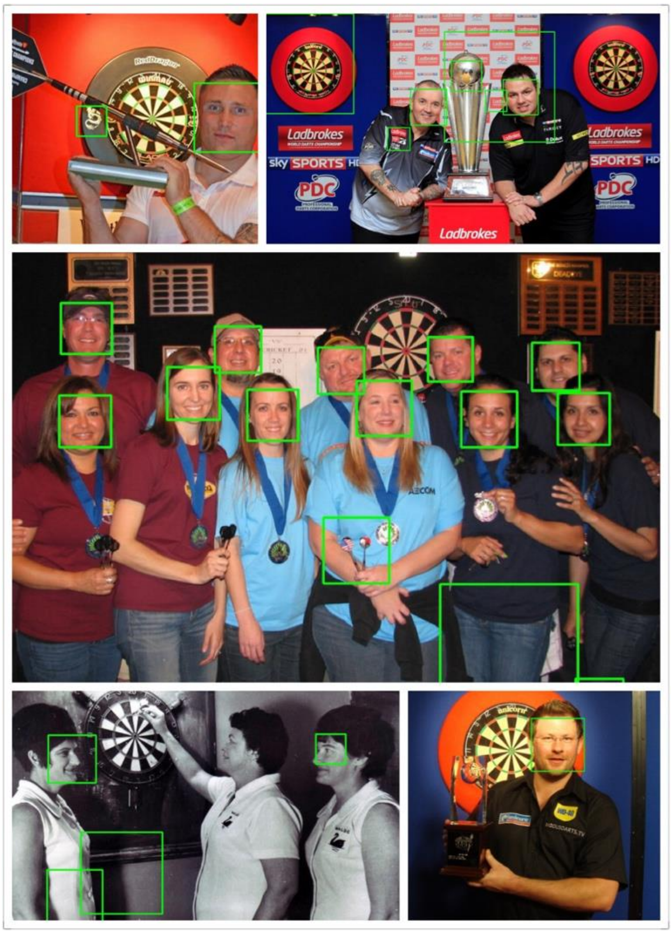

Figure 1—The Viola-Jones detector for face classifier (Top-left dart13.jpg, Top-right dart14.jpg, Middle dart5.jpg, Bottom-left dart15.jpg,Bottom-right dart4.jpg.

Detecting and locating instances of an object in images is important for many computer applications and an ongoing area of research. Also, it is a challenging task for a computer. Problems including the different colors of skin, varied facial expressions, different face orientation and lights could be able to cause some facial features deformed, which have a great impact on the accuracy of face judgement. team members.

As shown in the dart5.jpg of Figure 1, all faces in this image have been successfully recognized, thus the true positive rate(TPR) is 1. In dart15.jpg, two of three were identified so the TPR is 0.67.

However, only considering true positive rate cannot achieve a high performance for the Viola-Jones detector. As seen from dart5.jpg, each face has been identified but there are two non-face bounding boxes. Therefore, a better detector should reach a 100% TPR and a low false positive rate(FPR).

As for the reason why it is always possible to achieve a TPR of 100% on any detection task is that the way would identify every part of image to ensure that every object is detected, but that results in high FPR due to that non-object class are being incorrectly detected. In order to achieve high accuracy of object detection, we could train different types of features separately. For example, we could train different parts of face separately, also adjusting corresponding parameters is useful to some extent.

In our project, in order to calculate the F1-score, we store pixels of the faces in each image in csv document. When running the project, the faces would be bounded with red boxes. As you can see in Figure 2, we compute the area of overlap between ground truth box and each bounding box. When the area is over 50%, we regard that the face is detected correctly.

Figure 2—an example, how we compute ground truth with the red bounding box and green bounding box.

### Subtask2: Building & Testing your own Detector

The second task requires to build an object detector that recognizes dartboards. The training process is to use OpenCV's training tool to consider 1,000 positive images and 1,000 negative ones of dartboards. And the tool creates a strong classifier based on the Haar-like features of a dartboard. As shown in Figure 3, we plot the TPR and FPR at different stages of the training process.

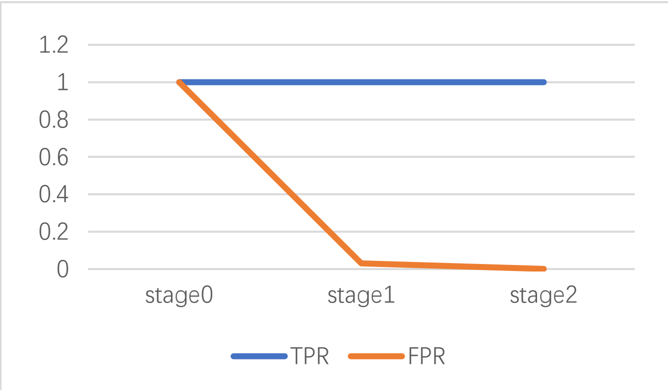


Figure 3 - A graph showing the changes of TPR and FPR of three stages.

Figure 3 shows that at each stage the TPR remains constantly at 1, whereas the FPR decrease from 1 to 0.0008.


Figure 4 - Viola Jones dart detector applied to dart 0.jpg(Top left), dart 1.jpg(Top right), dart 2.jpg(Bottom left) and dart 3.jpg(Bottom right)

The overall F1 score across all 16 images is shown in Figure 5 as you can see as follows.

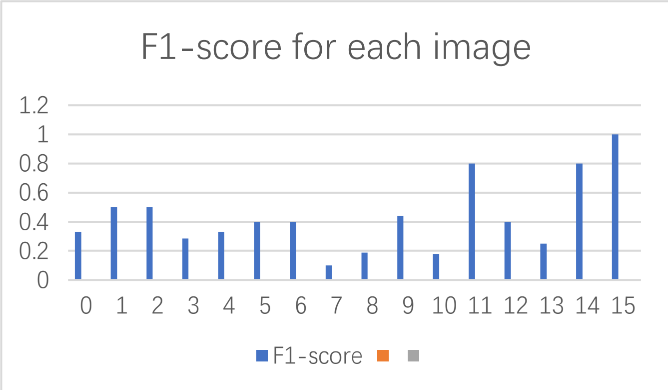


Figure 5—The overall F1 score across all 16 images.

Seen from Figure 5, F1 score for each image is not very high. Even though in the training process the FPR decrease, which might be caused by that the detector has over-fitted the training data, the performance of the detector on the test images is poor. As shown in Figure 4 and Figure 5, although all dartboards are recognized in each image, there are many cases where the detector identified a dartboard mistakenly.

In Figure 4, all dartboards have been identified but some non-dartboard areas have been identified as dartboard too, which results in its high false positive rate and low F1 score.

The poor performance of the detector in the real world might be caused by a small sample of negative training instances. In order to decrease its FPR on the test images, much more negative images would be added to the training process. Moreover, other methods should be implemented to detect the object correctly.

### Subtask 3: Integration with Shape Detectors

The third subtask required us to combine Hough Transform with our detector trained by Viola-Jones framework. Obviously, a dart contains a concentric circle and 20 diagonal lines intersecting at the center point. Therefore, we decided to apply **Hough Circles and Lines transform** to our detector generated in Subtask 2 to improve the performance of it. The flow chart of combining Hough transform and Viola-Jones dart detector is below.

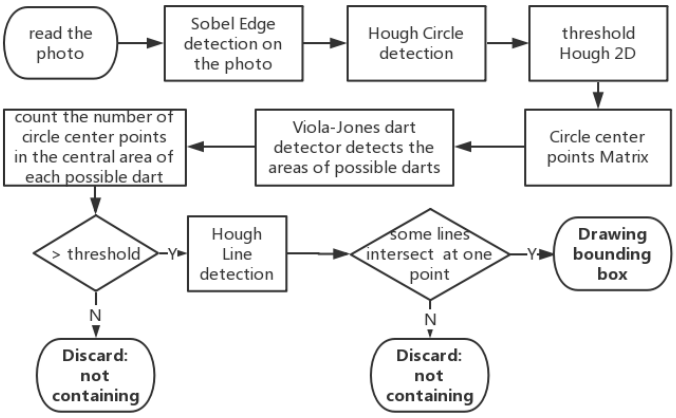


Figure 6 – the flow chart showing how to combine Hough Transform and Viola-Jones dart detector

As shown in Figure 6, we chose to calculate Hough Space over the whole photo rather than a single possible dart area when we did Hough Circles detection, because we observed that the possible dart areas detected by Viola-Jones dart detector sometimes cannot enclose the outer of real darts.

However, only each possible dart area is calculated Hough Space during the process of Hough Lines detection. This method can reduce the interference of the noise in spaces where there are many line structures outside the area darts. The results of detection can be seen in Table 1.

| TP | FN | FP | Precision | Recall | F1 |
|---|---|---|---|---|---|
| 21 | 3 | 2 | 0.913 | 0.875 | 0.894 |

Table1 – the result of our detector improved by Hough detection

As shown in Table 1, our model can get a very high score in detecting darts on the example images. Only 3 actual darts cannot be detected, and that results in our model has a high TPR (0.875). But there are still 2 FP darts (not actual darts are detected), therefore reducing FP is our purpose in subtask 4.

Figure 7 – dart4.jpg: Threshold gradient magnitude image (Top left); Hough Space 2D (Top right); Circle Center Points (Bottom left); Result image (Bottom right)

However, this model cannot detect all actual darts. In Figure 7, a small dart on the top of the trophy is not detected, because its outer is too fuzzy in edge detection.

Figure 8 – dart6.jpg: Threshold gradient magnitude image (Top left); Hough Space 2D (Top right); Circle Center Points (Bottom left); Result image (Bottom right)

To sum up, our model performs very well on most example images, like Figure 8, since darts in these images are very clear, and the Hough Circles detection can locate the circle center points accurately.

### Subtask 4: Improving your Detector

As mentioned in Subtask 3, the main mission in this task is to **reduce the number of FP darts**. To achieve this reduction, we decided to apply SURF (Speeded-Up Robust Features) to our detector.

SURF is a local feature detector and descriptor, which can be used for tasks such as object recognition, image registration etc. The process of using SURF to improve our dart detector is below.

*Step1: Using SURF Detector detect the keypoints of the source object image (a compressed image of "dart.bmp") and the target scene image*

*Step2: Calculate descriptors*

*Step3: Matching descriptor*

*Step4: Find the "good" matches (set a threshold for distance)*

*Step5: Find the transform between matched keypoints and do perspective transform to locate the corner points of target image*

*Step6: Draw the bounding box depend on the corner points*

Combining this bounding box with the possible dart areas from our detector can make the result more accurate.

However, due to the limitation of this method (explaining latter), we designed a vote-system to identify the "true" darts. Each possible dart area will be voted by the results of Hough Circles, Lines and Surf, and an area will be classified when it got 2 or 3 votes.

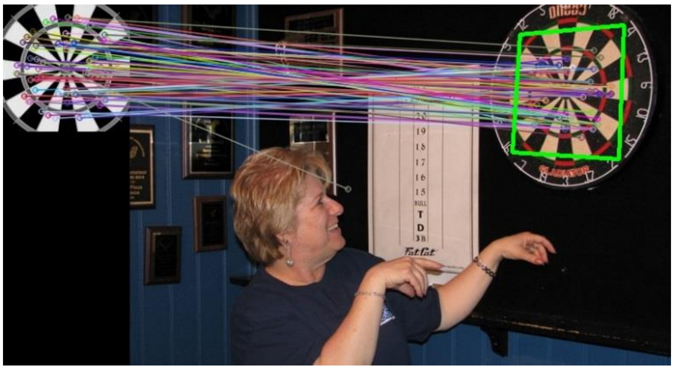

Figure 9 – Feature matching between "dart.bmp" and "dart0.jpg"

Although SURF detector sometimes performs very well (Figure 9), the Figure 10 and Figure 11 show the shortcoming of it. This detector can only detect one object in the target image. In other words, if the image has more than one dart, this detector cannot work well (Figure 10). In the Figure 11, apart from the area of real darts in the target image, the SURF also detects many other matching points. But the feature detector is unable to get the right area from scattered points.


Figure 10 – Feature matching between "dart.bmp" and "dart14.jpg"

Figure 11 – Feature matching between "dart.bmp" and "dart8.jpg"

Despite these limitations, our final detector also help us to reduce the FP from 2 to 0, thus, increasing the F1-score from 0.894 to 0.933. Although it looks like a small step, it is difficult to improve when the original detector works well. The final results are shown in Table 2 below.

| TP | FN | FP | Precision | Recall | F1 |
|---|---|---|---|---|---|
| 21 | 3 | 0 | 1 | 0.875 | 0.933 |

Table 2 – the result of our detector improved by SURF detector