Image Processing and Computer Vision Coursework

Themis Papathemistocleous
tp16381@my.bristol.ac.uk
Computer Science

Konstantina Psoma
kp16340@my.bristol.ac.uk
Computer Science with Innovation

December 2018

1 The Viola-Jones Object Detector

The first step for tackling this task was to annotate the test images and generate the ground truths, which are represented with red bounding boxes drawn around the faces. The resulting images with the detector's green bounding boxes drawn around the detected areas and the ground truth are shown below.

Figure 1: Viola-Jones face detection implemented on image dart4.jpg

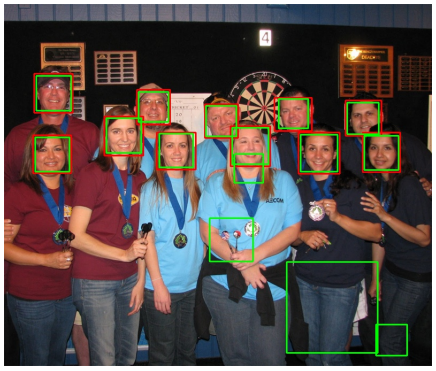

Figure 2: Viola-Jones face detection implemented on image dart5.jpg

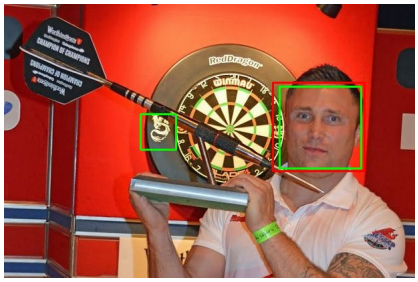

Figure 3: Viola-Jones face detection implemented on image dart13.jpg

Figure 4: Viola-Jones face detection implemented on image dart14.jpg

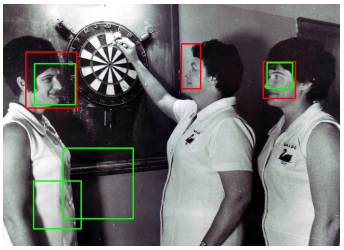

Figure 5: Viola-Jones face detection implemented on image dart15.jpg

It is interesting to observe how many of the faces were successfully detected by running the Viola-Jones's object detection. This can be expressed mathematically by calculating the True Positive Rate (TPR) of the images. For example, the image **dart5.jpg** has TPR $= \frac{11}{11} = 1$ and the image **dart15.jpg** has TPR $= \frac{2}{3} = 0.67$.

It is the case that there are practical difficulties in assessing the TPR accurately. An example of that can be depicted in image **dart15.jpg**, where the detection has drawn a box that covers half of the face of the rightmost person, and so it is difficult to decide whether the program has correctly detected and classified the object as a face, or not.

It is interesting to notice that it is always possible to a achieve a TPR of 100% with a very low threshold, so that the detector becomes very sensitive. In that case scenario, our image will be filled with boxes and some of them will be validating the ground truth. In order to better comprehend this concept, let's think of an example. Let's say that our True Positive is sick people correctly identified as sick. Then, we can easily achieve a TPR of 100% by identifying all people as sick.

For our program, we implemented a small piece of code to calculate the F_1-score of our face detection system accurately from our ground truth. If the detector's bounding box is inside the ground truth's box, then we consider the specific detection a True Positive, otherwise a False Positive. That way, we calculate the F_1-score using the formula:

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

, where $precision = \frac{TP}{P}$ and $recall = \frac{TP}{TP+FN}$.

A simple way to improve this technique would be to check what percentage of the ground truth the detection covers, and set a threshold for that value. That way, we can avoid considering as TP very small detections that are ambiguous.

2 Building and Testing Dart Detector

2.1 Training

Using the provided readily compiled versions of create samples and train cascade, we created 500 samples of the dart image and trained a dartboard detector via boosting. The training process was separated in 3 stages, each with an additional feature added to improve the detector. At each stage the TPR was 1, but the False Positive Rate (FPR), started from 1 and decreased in every following stage as shown in **Figure 6**. The value of FPR represents how many detected regions were not correct and since it decreases on every stage, that is proof that the detector improved in the sense that it has a very low rate of detecting an object that is invalid.

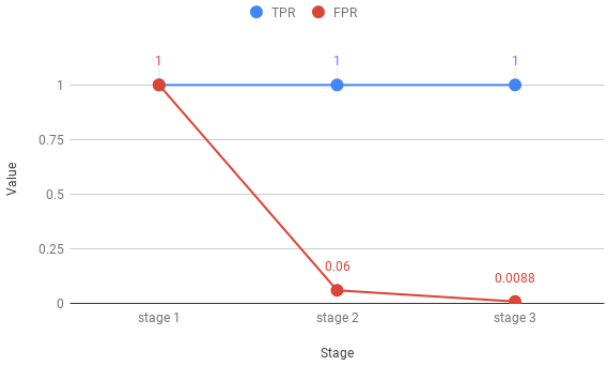

Figure 6: Graph plot of TPR and FPR at each of the three stages of training the dart Detector.

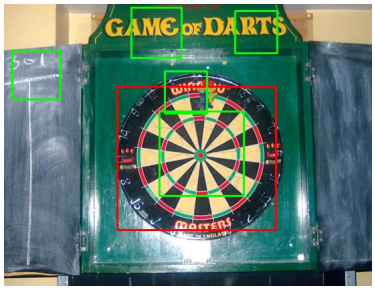

Figure 7: Viola-Jones dart detection implemented on image dart1.jpg

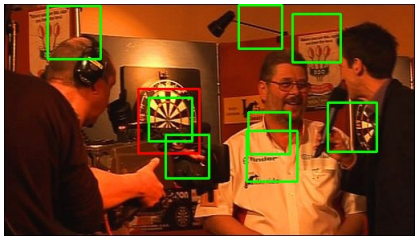

Figure 8: Viola-Jones dart detection implemented on image dart11.jpg

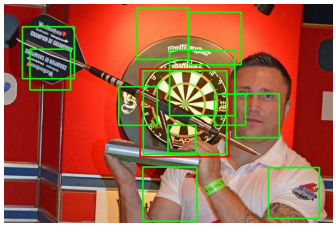

Figure 9: Viola-Jones dart detection implemented on image dart13.jpg

Figure 10: Viola-Jones dart detection implemented on image dart15.jpg

2.2 Testing

After acquiring the trained detector, we used it to detect dart boards on the example images, some of them are shown in **Figures 7, 8, 9 and 10**. The F_1-score of each image is calculated and depicted in the table below. The F_1-score table evaluates the accuracy of the detection algorithm implemented on each image as well as summing the program's performance by averaging the detection's accuracy over all 16 images. The results show that the detector did not achieve a very high accuracy and there were cases where the dartboard was not even detected. An example of the latter can be observed on our table, where the F_1-score for image dart12 is zero. The poor accuracy results were due to the fact that the FPR was particularly high. For example, in **Figure 9** the FPR was 11/12 which is much higher than the third stage's FPR of 0.0088 (see **Figure 6**). However, the average TPR over all the images remained relatively high with most of the dart boards successfully detected.

Image	**F_1-score**
dart0	0.25
dart1	0.30
dart2	0.20
dart3	0.29
dart4	0.20
dart5	0.14
dart6	0.00
dart7	0.20
dart8	0.18
dart9	0.29
dart10	0.33
dart11	0.22
dart12	0.00
dart13	0.15
dart14	0.10
dart15	0.67
Average	**0.23**

F_1-score table of Viola-Jones Detector

3 Integration with Line Detector

3.1 Implementation of Hough Line Detector

To extend the Viola-Jones detector and improve our results, we implemented a Hough Space Line Detector. Instead of taking into consideration the whole image for our Hough Space, we used the resulting detected boxes from the Viola-Jones to find the threshold magnitude image of each box (see **Figures 11 and 14**). Then we implemented the Hough Transform for Line detection in each of the boxes (see **Figures 12 and 15**). We set the threshold to 70 so that only the lines with the higher votes are considered, and then we set as a condition that there are at least 10 lines in each of the detected boxes. The boxes that fulfilled the conditions above were classified as valid detections, and were kept in the image (see **Figure 13**).

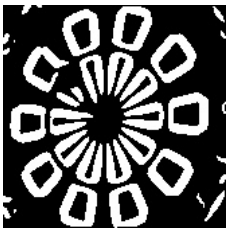

Figure 11: Magnitude image of the detected square containing the dartboard in image dart0.jpg.

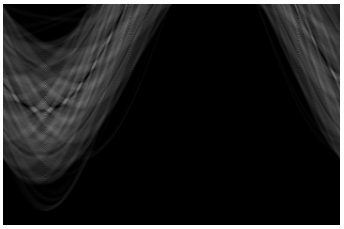

Figure 12: Hough Space for line detection of the detected square containing the dartboard in image dart0.jpg.

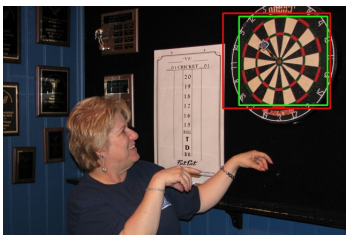

Figure 13: Improved detection with the implementation of the Hough Line Detector on image dart0.jpg.

Figure 14: Magnitude image of the detected square containing the dartboard in image dart2.jpg.

Figure 15: Hough Space for line detection of the detected square containing the dartboard in image dart2.jpg.

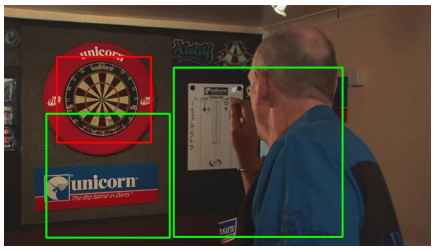

Figure 16: Detection of dartboard failed with Hough Line Detector in image dart2.jpg

3.2 Merits and Limitations

There is a noticeable increase in the average F_1-score of the table below, compared to the average F_1-score of the Viola-Jones detection alone. However, although in most images our improved detector managed to have a very high accuracy by achieving high TPR and very low FPR, such as the one shown in **Figure 13**, in some images it sabotaged the results by discarding previously correctly detected dartboards. An example of the latter is shown in **Figure 16**.

The merit of our implementation is that we managed to lower the FPR in our detection results. However, this resulted to a limitation because of the strict threshold of 70 implemented for our Hough Space Line Detector. The TPR is increased in images such as **Figure 16** but, even though the magnitude threshold shown in **Figure 14** is good enough and capable to detect the lines, our threshold is too high and therefore needs an even clearer magnitude image to validate it as a dartboard.

We did some experimenting, with the threshold values and observed that when we set the value to 60 for example, on the same image, **Figure 16**, the dartboard was successfully detected but we had an additional 3 False Positive detections.

Another limitation of our detector is that it only considers the rectangles resulting from Viola-Jones detection, which means that if Viola-Jones does not detect a dartboard in the first place then our detector will not consider that area of the image.

Image	F_1-score
dart0	1.00
dart1	0.67
dart2	0.00
dart3	0.00
dart4	0.67
dart5	0.00
dart6	0.00
dart7	0.67
dart8	0.22
dart9	0.50
dart10	0.85
dart11	0.00
dart12	0.00
dart13	0.40
dart14	0.25
dart15	1.00
Average	**0.40**

F_1-score table of our Line Detector

3.3 Rationale

Our Flow Diagram, **Figure 17**, shows the way that we combined Viola-Jones and Hough Transform. The main reason for combining the two in such way is that the Viola-Jones detector can be "relaxed" so that it has TPR equals 1 at all times and therefore the only areas of the image a Hough Transform needs to occur are only in the detected boxes from Viola-Jones. Using the Hough Transform for Line Detection helped decrease the FPR as much as possible, and get quite accurate results.

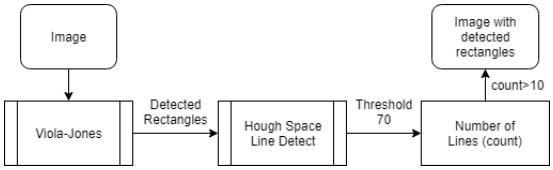

Figure 17: Flow diagram showing the combination of Viola-Jones and Hough Transform for Line Detection.

4 Improving our Detector

4.1 Rationale

To improve our detector further, we added our implementation of Hough Transform Circle Detector. As mentioned previously, if we "relax" our Line Detector threshold by setting its value to 60 instead of 70, we get a higher TPR. In our case, that is desirable even if it means a higher FPR as well.

The idea is that by implementing a Hough Circle Detector on the results of the "relaxed" version of our Line Detector, we will help decrease the FPR without affecting the TPR as much.

4.2 Visual Examples

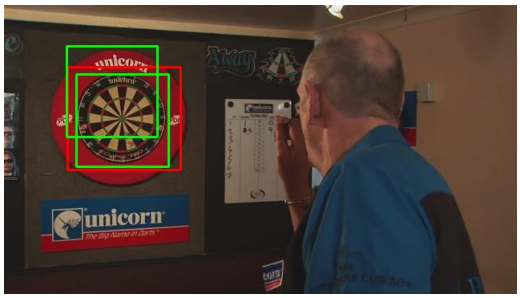

Figure 18: Successful detection of dartboard with implementation of Hough Circles and Hough Line Detector for the image dart2.jpg. It is worth noticing that its F_1-score increased from 0.00 to 0.67.

Figure 19: Huge improvement in the detection of the dartboard of image dart5.jpg with our improved Detector. Its F_1-score increased form 0.00 to 1.00!

4.3 Evaluation

Referencing the table below with the F_1-score of each image for our advanced Detector, a significant improvement is noticeable on the average performance of the Detectors, starting from 0.23 with Viola-Jones alone, to 0.40 with our Line Detector, and now up to 0.59.

By lowering the threshold of our Line Detector, we managed to recover some True Positive detected dartboards that were not considered before, such as the ones shown in **Figures 18 and 19**. For example, in **Figure 18** there were 2 detected areas, both False Negatives. Using our improved implementation, our result is now 2 detected areas with one of them being a TP, containing the dartboard.

We can observe, that our final detector is able to decrease the number of FN detected areas significantly. However, in the case of a steep camera angle, the circles of the dartboards look like ellipses, and as a result there is a chance they will not be detected.

There was only one case that the above fact affected negatively our results. The image dart10.jpg containing 3 dartboards, had them all detected with our Line Detect algorithm, while after combining the results with our Circle Detector in those areas only 1 of the dartboards was marked as valid, resulting a reduction in this image's F_1-score.

Image	F_1-score
dart0	1.00
dart1	1.00
dart2	0.67
dart3	0.00
dart4	0.67
dart5	1.00
dart6	0.00
dart7	0.67
dart8	0.50
dart9	1.00
dart10	0.50
dart11	0.00
dart12	0.00
dart13	0.67
dart14	0.80
dart15	1.00
Average	**0.59**

F_1-score table of our improved Detector

References

1. 'F1 score' (2018) Wikipedia. Available at `https://en.wikipedia.org/wiki/F1_score`

2. 'Record XY mouse coordinates on an uploaded image'. Mobilefish. Available at: `https://www.mobilefish.com/services/record_mouse_coordinates/record_mouse_coordinates.php`

3. 'cv::Rect Class Template Reference'. (2015) OpenCV documentation. Available at: `https://docs.opencv.org/3.1.0/d2/d44/classcv_1_1Rect__.html`