IMAGE PROCESSING & COMPUTER VISION

COMS30121, COURSEWORK 1

Farrel Zulkarnaen, Kheeran Naidu
fz16336, kn16063

Monday 10th December, 2018

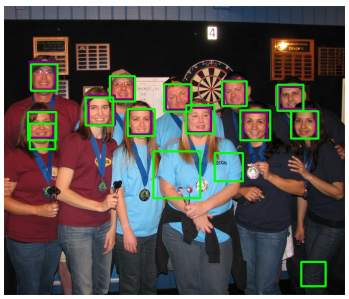
1 Viola-Jones Object Detector

Object detection is one of most sought after applications in computer vision, as it allows us to automate the process of monitoring objects. The most common example of object detection is those of faces, where the Viola-Jones object detector is the first framework that can detect faces in real time. This report will show a brief implementation of object detection using the Viola-Jones framework. [1]

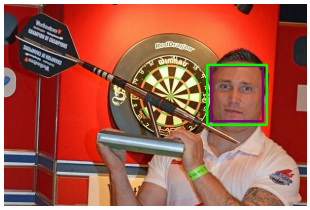
1.1 Part A: Implementation

By applying the provided classifier which has been trained for frontal faces, we can detect faces as seen in 5 example images in Figure 1 and examine the performance of the detector. First, we annotate the faces in the images with purple boundary boxes drawn around the faces as ground truth. Then, we will apply our classifier to detect the faces with their detection boundaries in green as shown in Figure 1.

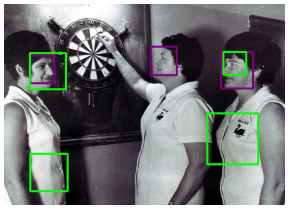
(a) dart4.jpg



(b) dart5.jpg



(c) dart13.jpg



(d) dart15.jpg

(e) dart14.jpg

Figure 1: Detected faces in the test images set: dart4.jpg, dart5.jpg, dart13.jpg, dart14.jpg, dart15.jpg. The purple boundary boxes are our groundtruth annotation whilst the green ones are detection by the Viola-Jones detector.

1.2 Part B: Evaluation

Take for examples Figures 1(b) and 1(d). In Figure 1(b), there are 11 faces in total, and the classifier detected all of the faces but also misclassified 4 instances giving a True Positive Rate (TPR) of 1 and a False Positive Rate (FPR) of 0.36. One instance of the misclassification is when the woman on the far left has been classified twice, with having a second face underneath her eyes. This classification most likely occurred because her cheeks are glistening due to lighting, causing a mistakenly classified eyes based on Haar-features standard (vertical gradient change on her cheeks is similar to those of eyes). As for Figure 1(d), the TPR for the image is 0.67, but the classifier failed to detect one face because we are using a test set based on frontal-viewed faces; giving us 1 false negative (FP). Facial features have greater variances when viewed from the sides, such as the infinitely many possible orientation for ears and eyes; compared to the constraint posed on faces viewed from the front. This shows the difficulties in gaining a 100% TPR, as explained in the case of Figure 1(d).

There are various uncertainties to account for in detecting faces which ends up posing practical difficulties in assessing TPR accurately. Firstly, variation in angle, scale, and lighting can skew detection for different faces. Secondly, when annotating images and setting ground truths we make subjective judgements based on where a face begins and ends, and this varies from person to person. As a result, this would add further uncertainty as to what determines an accurate true positive (TP) classification since the green detection boundaries might not precisely match the subjectively drawn purple boundaries. We solved this by evaluating the Harmonic Mean (H) of the proportion of each boundaries' overlapping area, and classifying it as a TP detection if H is above a threshold, which we set as 0.5.

On the on the other hand, as shown in the case of Figure 1(b) it is possible to achieve a 100% TPR on any detection task. If we tune our classifier to naively classify everything, by setting the minimum neighbour parameter to 0 [1], we can get a perfect TPR; albeit at the cost of many false positives. Therefore, we use an $F1$-score as a suitable evaluation measure for in-balance data because we need to marginalise our evaluation by taking into account both false positives and false negatives. As such, the $F1$-score for Figures 1(b) and 1(e) are 0.85 and 0.57 respectively, where an $F1$-score is given by the equation 1, which is the harmonic mean of precision and recall.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{1}$$

where [2]

$$Precision = \frac{TP}{TP + FP} \quad \text{and} \quad Recall = \frac{TP}{TP + FN} = TPR \tag{2}$$

[1] The *minimum neighbour* parameter specifies the number of neighbours each face candidate should have to retain it. Having it set as 0 will retain all face candidates naively, whereas a higher value refines the detection.

[2] FN is the number of false negatives.

2 Our Own Detector

2.1 Part A: Training

Besides faces, the Viola-Jones framework can also be trained to detect other objects; whereby for our case we will now be detecting dartboards. The Viola-Jones framework consist of 4 components: feature extraction of Haar-like features, usage of Image Integrals and usage of Adaboosting followed by filtering through a Cascade of classifiers. [1]

We utilise this framework to develop a new classifier that is trained for dartboards instead of faces. To do so, we create a training set consisting of both positive and negative samples; whereby the positive samples are images with dartboards present, and negative samples are images without dartboards. Furthermore, each of the sample contains images varying in angle and contrast, which was lacking for the training for frontal faces as explained previously.

In the dartboard training process, we get the results in Figure 2. From stage to stage, the True Positive Rate (TPR) remains 1, and so it consistently detected all present dartboards; however, the False Positive Rate (FPR) decreases per stage, with the largest decrease happening between stage 0 and 1. This shows that the training in 3 stages classifies different features at different stages of the cascade. Based on the figure, stage 0 trains the classifier to positively detect dartboards, whereas features that correspond to the absence of a dartboard are trained mostly in stage 1 and improved slightly in stage 2.

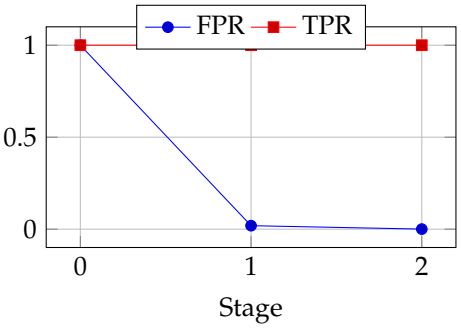
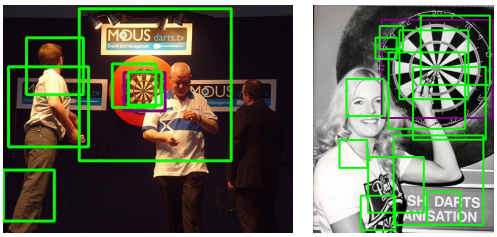


Figure 2: FPR vs TPR from stage to stage of the training process

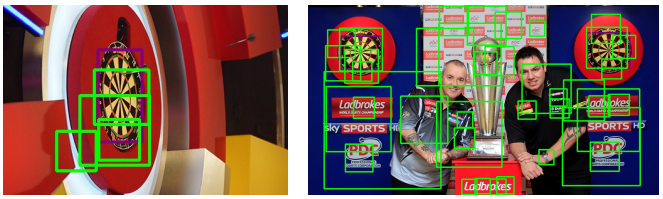
2.2 Part B: Comparison

After the training process, we apply the classifier to our image set. The images in Figure 3 shows the action of the classifier on 4 of the images. We first notice that there are many false positive classifications. This is contrary to our analysis in Part A, where we would expect a low false positive rate based on Figure 2. This tells us that the classifier has most likely been over-fitted to the training data, and so is imprecise with unseen data. This is no surprise as our positive samples were just the same image with different orientations and contrasts. In addition, as we set the minimum neighbour parameter to 1, only 1 neighbour is required for an object classification to be retained which then explains the large number of classifications in an area. As previously discussed we calculate the F1-score to determine the efficacy of the classifier.

As seen in Table 1, the F1-score for each image is very low, with an overall average (mean) of 0.190. F1-score is the harmonic mean of the precision and TPR, and so an ideal score would be 1.0. We see that the TPR of each and every image is 1.0, therefore the classifier correctly classifies the dartboards; however it's very imprecise with many false positives. This shows that the Haar-like features selected are ones that are simple and present in many other objects. In order to improve the classifier, we need to reduce the number of false positives.



(a) dart6.jpg (b) dart9.jpg



(c) dart12.jpg (d) dart14.jpg

Figure 3: 4 image result showing of our dartboard detection: (a) describes the first subfigure; (b) describes the second subfigure; (c) describes the third subfigure; and, (d) describes the last subfigure.

Images	F1-score	Images	F1-score
dart0	0.11	dart8	0.12
dart1	0.25	dart9	0.15
dart2	0.22	dart10	0.17
dart3	0.14	dart11	0.25
dart4	0.17	dart12	0.40
dart5	0.14	dart13	0.13
dart6	0.29	dart14	0.10
dart7	0.10	dart15	0.22
Mean		**0.19**	

Table 1: F1-scores for image set

Our evaluation method to determine if 2 detections (between classifications or ground truths) are equivalent is to calculate if the Harmonic Mean (Equation 3) of the overlapping areas with respect to each image is above our threshold=0.5. Figure 4 depicts this.

$$H = \frac{2 \times x \times y}{x + y} \tag{3}$$

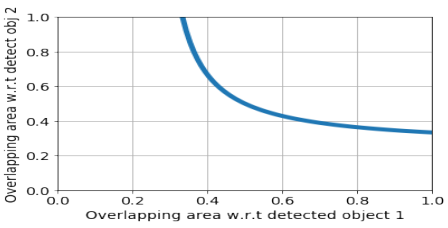


Figure 4: The area above the line contains the values of proportional overlapping area with respect to each image which is above our threshold = 0.5.

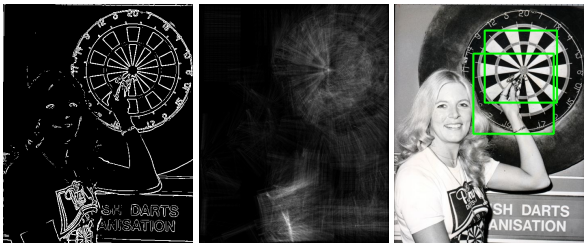
3 Integration With Shape Detectors

Apart from detecting objects based on features, a fundamental aspect of processing images is recognising objects from its *edges* and *shapes*. Since now we have defined a new object detector we can improve on it by integrating shape recognition via a *Hough Transform*, to reduce the false positives.

3.1 Part A: Hough Transform

We 'superimpose' the selected Haar-like features with the extracted shape features from the Hough Transform (HT) such that only the detection captured by both detectors would be retained, minimising false positives. Below are the results of applying the HT on images dart5.jpg and dart9.jpg, which illustrate the merits and limitations of our detector.

(a) edge5.jpg (b) houghspace5.jpg (c) output5.jpg



(d) edge9.jpg (e) houghspace9.jpg (f) output9.jpg

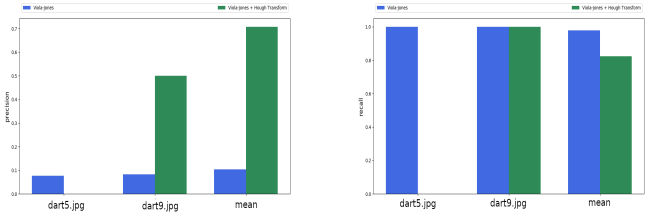
Figure 5: Thresholded gradient magnitude, 2D representation of the Hough-space, Resulting image with detection captured in green boundary boxes for dart14.jpg and dart14.jpg respectively.

3.2 Part B: Evaluation comparison

- The merit of our implementation is the increase in precision of our detection, depicted in Figure 6(a) by the overall average, this implies that most of the false positives (FP) captured by Viola-Jones (VJ) have now been discarded by combining the HT. As an example, refer to Figure 3(b) with Figure 5(f). Studying Figure 5(e) also shows that the FP on the woman's t-shirt has been discarded too.

- The limitation of our implementation is that since the VJ has a low minimum neighbour parameter, for every true positive classifications there are multiple FP nearby. This means that our superimposition with HT has a chance of outputting multiple classifications for the same object if in the VJ detection there are overlaying detection. This is depicted in Figure 5(f).

- We also note that in Figure 5(c) there is no dart board detection even though it is detected with VJ (as we have 100% TPR) and the Hough Space in Figure 5(b) clearly shows that the dart board is detected, depicted in Figure 6; this is because VJ detects the full diameter of the dartboard whereas HT detects the inner circle, so the evaluation is below the threshold.



(a) precision levels (b) recall levels

Figure 6: Bar chart depicting the differences between precision and recall of dart5.jpg, dart9.jpg and the overall average(mean) detection done by Viola-Jones (blue) and Viola-Jones with Hough Transform (green).

Note that the ideal result would be a average(mean) score of 1 for both precision and recall, this means that we seek to increase precision from before (minimise false positives) whilst retaining the high TPR we already had. Our results show close to what what we had hoped to achieve. To further evaluate our implementation, Table 2 below shows the new $F1$-scores, and if you compare it to Table 1 we now have a significant increase in mean $F1$-score from 0.19 to 0.73.

Images	F1-score	Images	F1-score
dart0	1.00	dart8	0.67
dart1	1.00	dart9	0.67
dart2	1.00	dart10	0.80
dart3	0.50	dart11	1.00
dart4	1.00	dart12	0.00
dart5	0.00	dart13	0.50
dart6	1.00	dart14	0.80
dart7	1.00	dart15	0.67
Mean		**0.73**	

Table 2: F1-scores for image set with shape recognition

3.3 Part C: Flow chart

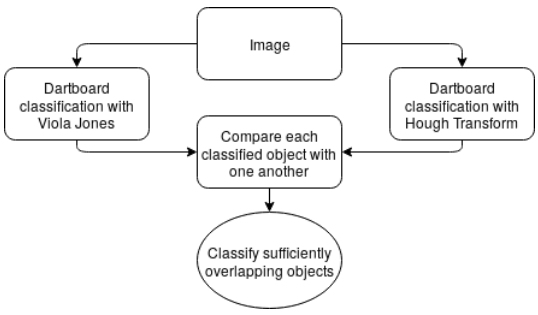


Figure 7: Flow diagram showing how we merged VJ and HT.

- VJ classifies based on Haar-like features, whereas HT classifies based on shapes, circles in this case. Both have high TPR and so when overlapped it will still detect a dart board correctly, retaining the high recall.

- Both search for very different features and so their false positives are very unlikely to be the same, therefore an overlap removes most, if not all, the false positives and increasing the precision in the process.

4 Improvement

All in all, there are drawbacks to the procedures and implementations that we have done so far. First, by having a low minimum neighbour parameter, we naively aimed to maximise TPR whilst hoping that the Hough-Transform would 'filter' out the remaining false positives. While this does improve the precision, it does not solve the problem of multiple classification due to an overlay of FP and TP. Second, another problem lies in the lack of training samples. To better perform object detection it is desirable to incorporate a diverse range of training images. Here are possible ways to improve on the drawbacks stated above.

To improve on the efficacy of our detection we tuned 3 parameters which we noticed to greatly impact our result; these parameters are the minimum neighbour and minimum size parameter for Viola-Jones' cascade classifier, and the edge threshold for Hough Transform.

- By increasing the minimum neighbour parameter we are able to retained less false positives in the Viola-Jones detection stage of our implementation. Therefore, with less overlay chances between false positives and true positives we reduced the possibility of multiple classification as seen in Figure 5(f). Here we set it to 10.
- In addition, the minimum size parameter defines the smallest scale for our object that is allowed by our detector. We noticed that in one of the images, the dart size is smaller that our predefined setting, so we decrease it to equal to size of the images created in the training stage of our classifier i.e 20x20.
- Finally, to fine tune the Hough Transform (HT), we set the gradient magnitude threshold limit slightly lower (to 2 times the mean gradient magnitude) to consider more edges in the image. We can do this now as VJ doesn't have many false positives anymore and so an increase in our false positives for HT is not as bad. This also increases the running time of the HT algorithm, but allows for dartboards in low lighting to be detected.

Additionally, we improved the efficiency of the Hough Transform algorithm by classifying our dartboards considering only the top left hand quarter of a circle, this reduced the computation 4 folds. The limitation of this is that if the top right hand quarter of the dartboard was blocked, then the entire dartboard wouldn't be considered.

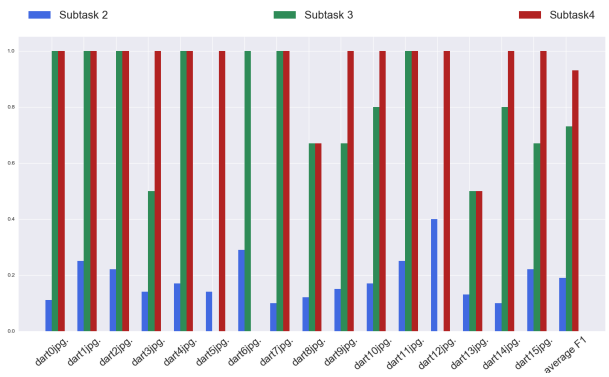


Figure 8: Comparison of $F1$ scores computed in subtask 2, 3, and 4. Labelled as blue, green and red respectively.

By simply tuning the parameter we have gained a 20% increase in average $F1$-score, if you refer to Tables 1 and 2, previously we had an average $F1$-scores of 0.19 and 0.73 for subtask 2 and 3 respectively. Now, with the tuned parameters we achieved an average $F1$-score of 0.93.

5 Future Works

This section briefly outlines further possible implementation that is beyond the scope of the assignment, yet still highly relevant and applicable. The main disadvantage of the Viola-Jones framework - which has been the main focus of this report - is that it is a decade old framework that operates with on sliding window model which can cause multiple classification for a single object with the wrong parameter choice as explained in our Improvement Section. Therefore, below we propose an alternatives to the Viola-Jones framework through the means of neural networks for future expansion.

Deep Learning Models

There are many examples of object detection algorithms. Traditionally, these algorithms are iterative methods of training a model, and so in order to test the successfully classification on each step we must evaluate the classifier with the ground truth. We do this using Intersection over Union [4] (IoU), which is very similar to the Harmonic mean of the proportion of overlapping area with respect to each image.

In most neural networks for object detection, whereby our output is a bounding box around the classified object, the training process for object detection, involves using the IoU as a loss function for updating the weights and biases of the neural network in back propagation of the training process, and it is from this process that the model can learn which detection box best fit our expectation; therefore, erasing the chances of multiple classification.

A simple neural network, the simplest perhaps, is a single layer feed forward neural network [5]. This, however, doesn't represent the complex biological image processing of the visual cortex studied by David Hubel and Torsten Wiesel [6]. Therefore, we use a much more representative convoluted neural network (CNN) for the task of object detection, which has many implementations [7]. In our case we focus on using one of the latest and most efficient implementations, Single-Shot Multibox Detection (SSD); with a latest release being tiny SSD, which is a much more efficient version of SSD [8].

In implementing SSD, we trained it using PASCAL Visual Object data sets (VOC2007 & VOC2012) [9]. When running the classifier on our test images, it detected every person with great precision, as seen in Figure 9. However, dartboards are not included in the PASCAL VISUAL Object data classes and also not in any data set readily available online. Therefore, we would need to create our own data set of dartboards in order to detect them using SSD. We have researched how to quickly build a custom image data sets using Microsoft Bing's image search API [10] and an annotation tool which saves the ground truth into an XML file in Pascal VOC format [11]. We would then be able to train our CNN and test it as a dartboard classifier. However, the limitation of this is that we would still need to annotate thousands of images manually. If we had the opportunity to do so, we would able to show the modern application of object detection by not only detecting dartboards in images, but also in live-video feeds.

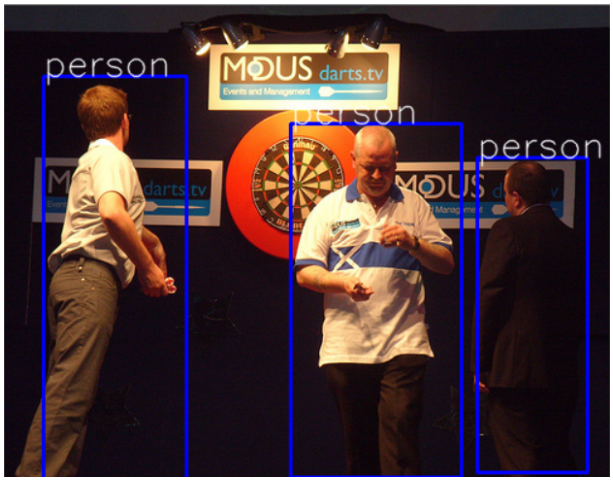


Figure 9: Classification result via SSD model for dart2.jpg using Pascal VOC2012 dataset.

References

[1] Robust Real-time Object Detection, Paul Viola & Michael Jones, 2001 http://www.hpl.hp.com/techreports/Compaq-DEC/CRL-2001-1.pdf

[2] OpenCV - CascadeClassifier Class Reference https://docs.opencv.org/3.4/d1/de5/classcv_1_1CascadeClassifier.html

[3] Geometric & Harmonic Means in Data Analysis, Daniel McNichol, 2018 https://towardsdatascience.com/on-average-youre-using-the-wrong-average-\geometric-harmonic-means-in-data-analysis-2a703e21ea0

[4] Intersection over Union (IoU) for object detection, Adrian Rosebrock, 2016. https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

[5] Single-layer Neural Networks (Perceptrons), Dr. Mark Humphrys, School of Computing. Dublin City University. https://computing.dcu.ie/~humphrys/Notes/Neural/single.neural.html

[6] Hubel and Wiesel the Neural Basis of Visual Perception, Fehlhaber, Kate., 2014. https://knowingneurons.com/2014/10/29/hubel-and-wiesel-the-neural-basis-of-visual-perception/

[7] R-CNN, Fast R-CNN, Faster R-CNN, YOLOŁŁObject Detection Algorithms, Rohith Gandhi, 2018. https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection\-algorithms-36d53571365e

[8] A Tiny Single-shot Detection Deep Convolutional Neural Network for Real-time Embedded Object Detection, Alexander Wong, Mohammad Javad Shafiee, Francis Li, Brendan Chwyl, 2018. https://arxiv.org/pdf/1802.06488.pdf

[9] The PASCAL Visual Object Classes project, Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J. and Zisserman, 2005-2012. http://host.robots.ox.ac.uk/pascal/VOC/

[10] How to (quickly) build a deep learning image data set. Rosebrock, Adrian. 2018 https://www.pyimagesearch.com/2018/04/09/how-to-quickly-build-a-deep-learning-image-dataset/

[11] Graphical annotation tool using Qt. Tzutalin. LabelImg. Git code (2015). https://github.com/tzutalin/labelImg

[12] Histogram of Oriented Gradients and Object Detection. Rosebrock, Adrrian. 2018 https://www.pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection/