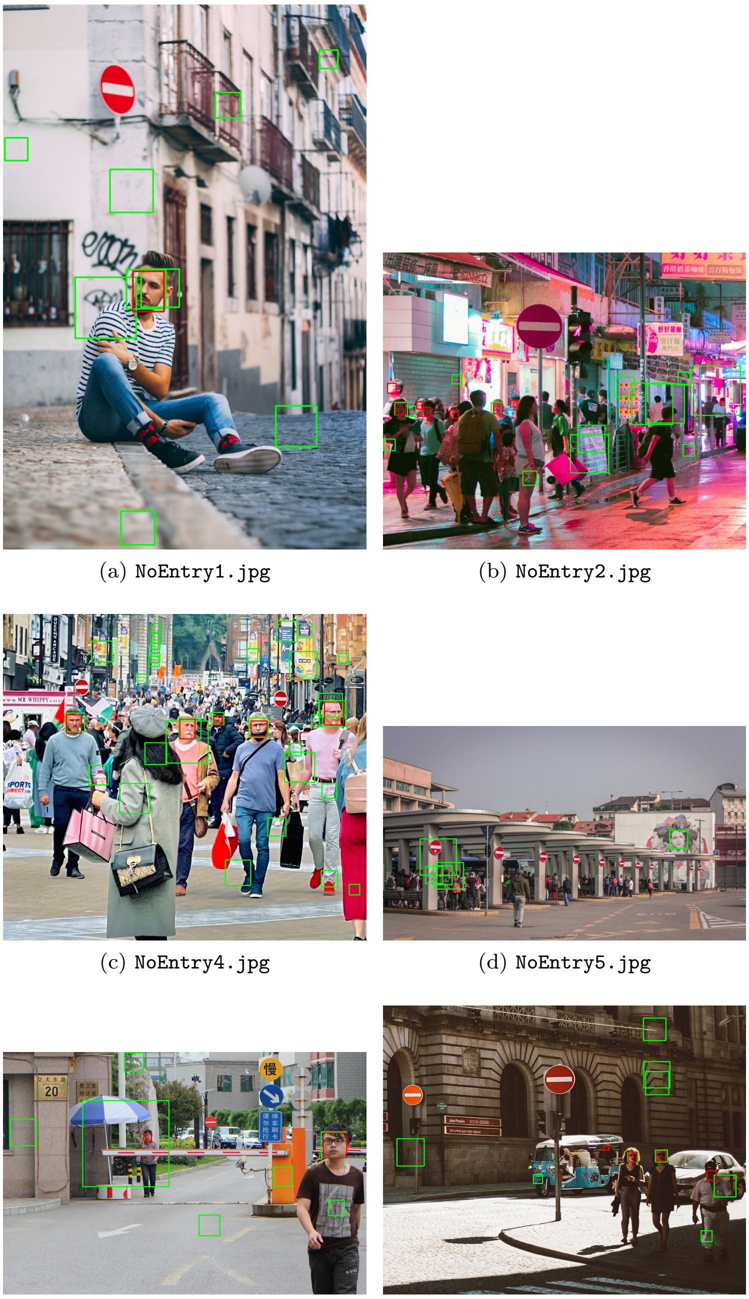
No Entry Sign Challenge Report

1 The Viola-Jones object detector

1.1 Ground truth and visualisation



(a) `NoEntry1.jpg`

(b) `NoEntry2.jpg`

(c) `NoEntry4.jpg`

(d) `NoEntry5.jpg`

(e) `NoEntry7.jpg`

(f) `NoEntry11.jpg`

Figure 1: Six images with the bounding boxes of the ground truths (in red) and actually detected instances (in green) from the frontal face detector

In an object detection task, ground truth bounding boxes assist in determining the accuracy of the detection algorithm, and help to visualise how well it performs. Figure 1 displays six images, with their ground truth bounding boxes in red, and the frontal faces detected by `face.cpp`, which implements the Viola-Jones object detection framework, in green.

1.2 Intersection-over-union, true positve rate and F_1 score

True positive rate (TPR) is a popular metric used to assess the performance of an object detection algorithm. TPR represents the probability that the object an algorithm is detecting will be positively detected, and is given by the formula:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

where TP is the number of true positives, and FN is the number of false negatives.

The first practical difficulty that arises in calculating the TPR is whether or not to define a predicted bounding box as a being a true positive or a false positive. I opted to define a detected bounding box as being a true positive if it had an intersection-over-union (IoU) value greater than 0.5 with a ground truth bounding box, as this is considered a good score [1]. For two bounding boxes A and B, the IoU is calculated as follows:

$$\text{IoU}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

where $|A \cap B|$ is the area of intersection, and $|A \cup B|$ is the area of union. Moreover, for each ground truth bounding box, I only defined the detected bounding box with the largest IoU value as being a true positive if there was more than one intersecting detected bounding box.

In any detection task, TPR can be a flawed metric to use to define how well a given model performs. This is because, firstly, the TPR can be usually be artificially increased simply by lowering the IoU threshold, without the model actually performing any better in reality. Secondly, it is possible to achieve a TPR of 100% by detecting every possible pixel region as being the object being detected, despite having a large number of false positives.

As a result of this, the F_1 score is often used as a measure of a model's accuracy. F_1 score is the harmonic mean of recall (i.e. TPR) and precision (i.e. the probability that a detected positive is actually positive), and is calculated as follows:

$$F_1 = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FP} + \text{FN})}$$

where FP is the number of false positives.

Table 1 displays the F_1 score and TPR that `face.cpp` achieved on each of the `NoEntry*.jpg` images. Due to the formulas for TPR and F_1 score, many of the values are undefined since division by zero yields an undefined result.

Table 1: TPR and F_1 score of the frontal face detector

Image	TPR	F_1 score
`NoEntry0.jpg`	Undefined	0.00
`NoEntry1.jpg`	1.00	0.20
`NoEntry2.jpg`	0.25	0.18
`NoEntry3.jpg`	Undefined	Undefined
`NoEntry4.jpg`	1.00	0.28
`NoEntry5.jpg`	Undefined	0.00
`NoEntry6.jpg`	Undefined	0.00
`NoEntry7.jpg`	0.50	0.22
`NoEntry8.jpg`	Undefined	0.00
`NoEntry9.jpg`	Undefined	0.00
`NoEntry10.jpg`	Undefined	0.00
`NoEntry11.jpg`	0.50	0.31
`NoEntry12.jpg`	0.00	0.00
`NoEntry13.jpg`	Undefined	0.00
`NoEntry14.jpg`	Undefined	0.00
`NoEntry15.jpg`	Undefined	0.00

2 Building and testing my own detector

2.1 Training performance

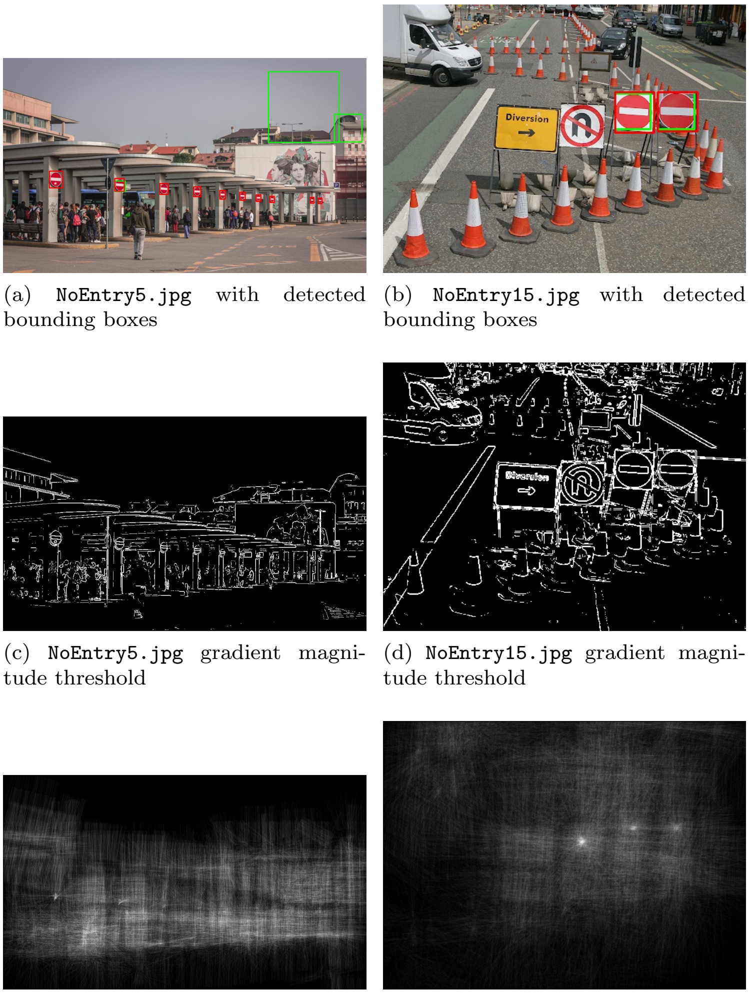
ROC graph

2.2 Testing performance

Table 2: TPR and F_1 score of Viola-Jones no entry sign detector

Image	TPR	F_1 score
`NoEntry0.jpg`	0.00	0.00
`NoEntry1.jpg`	1.00	0.13
`NoEntry2.jpg`	1.00	0.08
`NoEntry3.jpg`	0.50	0.15
`NoEntry4.jpg`	1.00	0.16
`NoEntry5.jpg`	0.30	0.17
`NoEntry6.jpg`	0.00	0.00
`NoEntry7.jpg`	0.00	0.00
`NoEntry8.jpg`	0.57	0.62
`NoEntry9.jpg`	0.00	0.00
`NoEntry10.jpg`	0.67	0.33
`NoEntry11.jpg`	0.50	0.11
`NoEntry12.jpg`	0.38	0.21
`NoEntry13.jpg`	0.00	0.00
`NoEntry14.jpg`	0.00	0.00
`NoEntry15.jpg`	0.50	0.40
All images	0.40	0.12

3 Integration with shape detectors


(a) NoEntry5.jpg with detected bounding boxes

(b) NoEntry15.jpg with detected bounding boxes

(c) NoEntry5.jpg gradient magnitude threshold

(d) NoEntry15.jpg gradient magnitude threshold

(e) NoEntry5.jpg 2D Hough space

(f) NoEntry15.jpg 2D Hough space

Figure 2: Two examples of no entry sign images which exhibit the merits and limitations of my implementation

3.1 Evaluation

Table 3: TPR and F_1 score of the Viola-Jones integrated with circle Hough transform no entry sign detector and difference with Viola-Jones no entry sign detector results

	Result		Difference	
Image	TPR	F_1 score	TPR	F_1 score
NoEntry0.jpg	0.50	0.67	+0.50	+0.67
NoEntry1.jpg	1.00	1.00	±0.00	+0.87
NoEntry2.jpg	1.00	1.00	±0.00	+0.92
NoEntry3.jpg	1.00	0.80	+0.50	+0.65
NoEntry4.jpg	1.00	1.00	±0.00	+0.84
NoEntry5.jpg	0.10	0.15	−0.20	−0.02
NoEntry6.jpg	0.50	0.67	+0.50	+0.67
NoEntry7.jpg	0.00	0.00	±0.00	±0.00
NoEntry8.jpg	0.43	0.60	−0.14	−0.02
NoEntry9.jpg	1.00	1.00	+1.00	+1.00
NoEntry10.jpg	0.67	0.80	±0.00	+0.47
NoEntry11.jpg	0.50	0.67	±0.00	+0.56
NoEntry12.jpg	0.13	0.22	−0.25	+0.01
NoEntry13.jpg	0.00	0.00	±0.00	±0.00
NoEntry14.jpg	0.00	0.00	±0.00	±0.00
NoEntry15.jpg	1.00	1.00	+0.50	+0.60
All images	0.42	0.56	+0.02	+0.45

Table 3 displays the TPR and F_1 score of this implementation, as well as the difference compared with the Viola-Jones implementation. The key merits and shortcomings include:

- A vastly improved F_1 score, as a result of only defining no entry signs detected by the Viola-Jones detector as positive if accompanied by a corresponding circle in the circle Hough transform
- No real improvement to the TPR score, as essentially no extra no entry signs can be detected due to the algorithm used to integrate the two detectors

3.2 Detection pipeline

Figure 3 outlines the way I combined evidence in my algorithm. The rationale behind this was as follows:

- The Viola-Jones detector detected approximately 40% of no entry signs, but had a large number of false positives
- Upon inspection of the detected bounding boxes, it became clear the Viola-Jones detector appeared to be detecting regions with a light bar horizontally between two dark bars
- Since no entry signs are circles, we could choose to detect the circles from the circle Hough transform that have an IoU with a Viola-Jones bounding box of greater than 50%

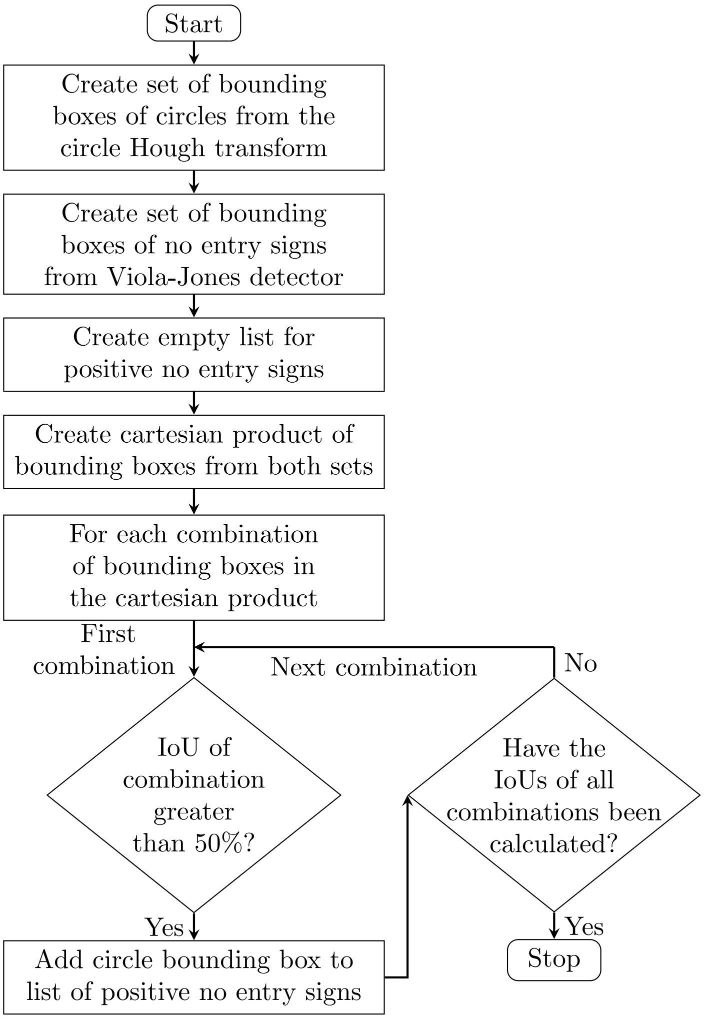

Start
Create set of bounding boxes of circles from the circle Hough transform
Create set of bounding boxes of no entry signs from Viola-Jones detector
Create empty list for positive no entry signs
Create cartesian product of bounding boxes from both sets
For each combination of bounding boxes in the cartesian product
First combination
Next combination
No
IoU of combination greater than 50%?
Have the IoUs of all combinations been calculated?
Yes
Yes
Add circle bounding box to list of positive no entry signs
Stop

Figure 3: Flow chart detailing my algorithm that integrates Viola-Jones with the circle Hough transform

4 Improving my detector

References

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