

No Entry Sign Challenge Report

1 The Viola-Jones object detector

1.1 Ground truth and visualisation



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 positive rate and F₁ score

True positive rate (TPR) is a popular metric used to assess the performance of an object detection algorithm. The TPR is the proportion of objects that an algorithm is attempting to detect

that are actually 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 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 F₁ score is often used as a measure of a model's accuracy. F₁ score is the harmonic mean of recall (i.e. TPR) and precision (i.e. the proportion of predicted positives that are 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: TPRs and F₁ scores 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

Table 1 displays the TPR and F₁ score that `face.cpp` achieved on each of the `NoEntry*.jpg` images. Due to the definitions of TPR and F₁ score, many of the values are undefined since division by zero yields an undefined result.

2 Building and testing my own detector

2.1 Training performance

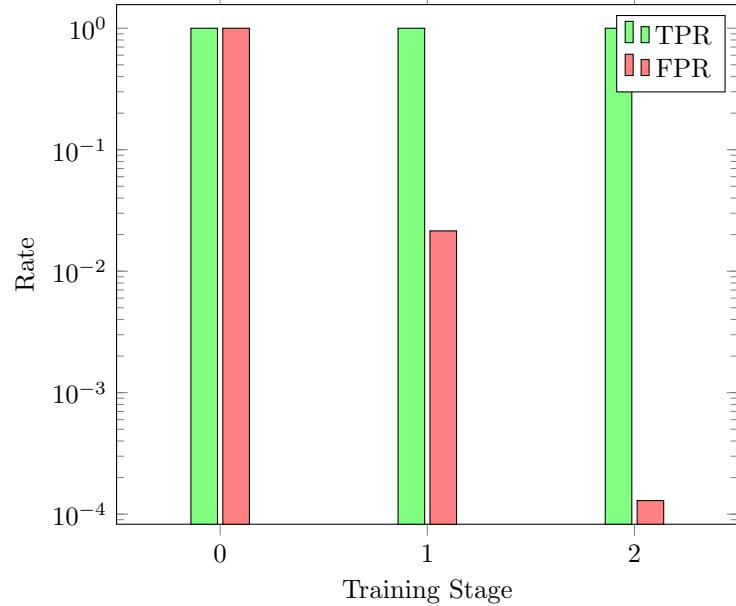


Figure 2: TPR and FPR of stages in the Viola-Jones no entry sign training process

During the Viola-Jones no entry sign training process, the false positive rate (FPR) is the number of negatives incorrectly predicted as being a positive no entry sign. Figure 2 displays the TPR and FPR at each stage of the training process. At each successive stage, the TPR remains at 100%, while the FPR decreases drastically.

2.2 Testing performance

Figure 3 contains three images with ground truth bounding boxes in red, and detected bounding boxes from the Viola-Jones implementation in green. The implementation works relatively well, achieving a TPR of 0.44 and an F_1 score of 0.48, as shown in Table 2. However, the implementation has its shortcomings. When tested, the Viola-Jones implementation detects a large number of false positives. Upon inspection, this primarily occurs in regions containing a slightly lighter strip horizontally between two darker strips. For example, the turn left sign in Figure 3c has been incorrectly detected in this manner.

The TPR on the test data is significantly lower than on the training data. One reason for this is likely due to the differing appearances of no entry signs in the test data. The Viola-Jones implementation was trained using a single no entry sign with different perspective transformations applied to it to create a collection of artificial positives. This approach works best for completely rigid objects [2]. However, due to the slightly differing appearances of the no entry signs between countries, the positives in our test images are not. Additionally, we are only testing our implementation on 16 images. To get a more accurate idea of the true TPR and F_1 score, we would need to test our implementation on a larger test set.



Figure 3: Three images with the bounding boxes of the ground truths in red and actually detected instances in green from the Viola-Jones no entry sign detector

Table 2: TPRs and F_1 scores of Viola-Jones no entry sign detector

Image	TPR	F_1 Score
NoEntry0.jpg	1.00	0.67
NoEntry1.jpg	1.00	0.40
NoEntry2.jpg	1.00	0.40
NoEntry3.jpg	1.00	0.80
NoEntry4.jpg	0.50	0.67
NoEntry5.jpg	0.40	0.47
NoEntry6.jpg	0.00	0.00
NoEntry7.jpg	0.00	0.00
NoEntry8.jpg	0.57	0.72
NoEntry9.jpg	0.00	0.00
NoEntry10.jpg	0.67	0.67
NoEntry11.jpg	0.50	0.25
NoEntry12.jpg	0.25	0.40
NoEntry13.jpg	0.00	0.00
NoEntry14.jpg	1.00	1.00
NoEntry15.jpg	0.50	0.67
All images	0.44	0.48

3 Integration with shape detectors

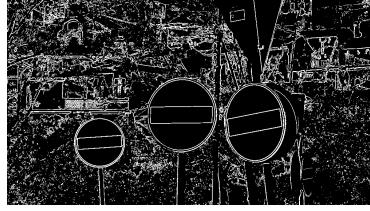
3.1 Hough Details



(a) NoEntry6.jpg with detected bounding boxes



(b) NoEntry2.jpg with detected bounding boxes



(c) NoEntry6.jpg gradient magnitude threshold



(d) NoEntry2.jpg gradient magnitude threshold



(e) NoEntry6.jpg 2D Hough space

(f) NoEntry2.jpg 2D Hough space

Figure 4: Two no entry sign images at various stages of the integrated implementation

3.2 Evaluation

Table 4 displays the TPR and F_1 score of the integrated implementation, as well as the differences compared with the Viola-Jones implementation. The key merits and shortcomings include:

- A significantly larger F_1 score, primarily as a result of increasing the precision; achieved by only defining an object detected by the Viola-Jones detector as positive if it has an IOU value with a bounding box from the circle Hough transform greater than 50%
- A small improvement to the TPR score, by reducing the minimum number of neighbours allowing for more true positives to be identified (the extra false positives were ‘filtered out’ by utilising the circle Hough transform)
- If a no entry sign is positively detected by one of the detectors, but not the other, it is not positively detected by the integrated implementation
- A significantly slower implementation, since the circle Hough transform takes longer to calculate

3.3 Detection pipeline

Figure 5 outlines the way I combined evidence in my algorithm. The rationale behind this was as follows:

- The Viola-Jones detector detected 44% of no entry signs, but had a low level of precision
- 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% (i.e. they are likely detecting the same object)

Table 3: TPRs and F_1 scores of the integrated implementation, and the differences compared to the Viola-Jones implementation

Image	Result		Difference	
	TPR	F_1 Score	TPR	F_1 Score
NoEntry0.jpg	1.00	1.00	± 0.00	+0.33
NoEntry1.jpg	1.00	1.00	± 0.00	+0.60
NoEntry2.jpg	1.00	1.00	± 0.00	+0.60
NoEntry3.jpg	1.00	1.00	± 0.00	+0.20
NoEntry4.jpg	1.00	1.00	+0.50	+0.33
NoEntry5.jpg	0.20	0.31	-0.20	-0.16
NoEntry6.jpg	0.00	0.00	± 0.00	± 0.00
NoEntry7.jpg	0.00	0.00	± 0.00	± 0.00
NoEntry8.jpg	0.43	0.60	-0.14	-0.13
NoEntry9.jpg	1.00	1.00	+1.00	+1.00
NoEntry10.jpg	1.00	1.00	+0.33	+0.33
NoEntry11.jpg	0.50	0.67	± 0.00	+0.42
NoEntry12.jpg	0.38	0.55	+0.13	+0.16
NoEntry13.jpg	0.00	0.00	± 0.00	± 0.00
NoEntry14.jpg	1.00	1.00	± 0.00	± 0.00
NoEntry15.jpg	1.00	1.00	+0.50	+0.33
All images	0.50	0.66	+0.04	+0.17

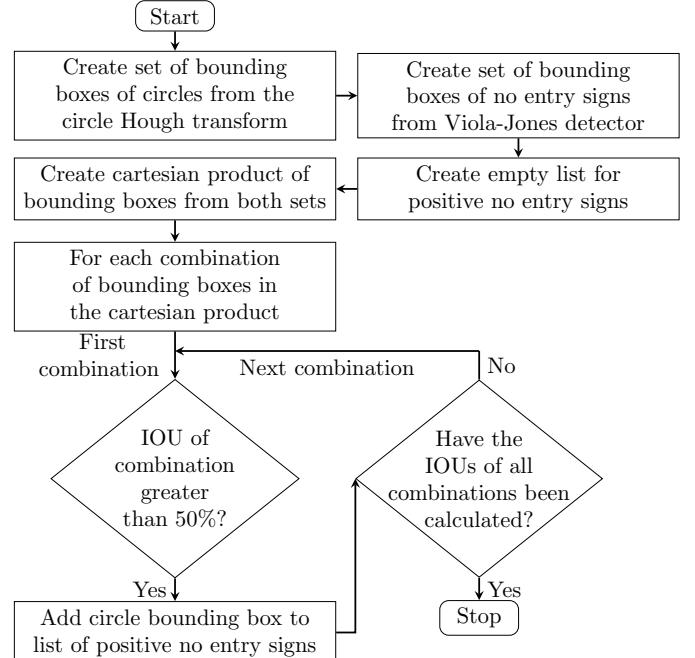


Figure 5: Flow chart detailing my algorithm that integrates Viola-Jones with the circle Hough transform

4 Improving my detector

4.1 Idea

The motivation behind the new approach came from identifying the areas whereby the previous approach failed. There were several no entry signs that were visually quite clear (i.e. big, no occlusion, etc.), and were identified by the circle Hough transform, but since the Viola-Jones detector did not identify them, they were not positively identified. Therefore, to improve the detector, I took every circle rejected by the previous approach, and put them through an extra stage of processing.

The extra stage of processing worked on the principal that every no entry sign shares similar characteristics. They contain two colours: red and white; and they have a horizontal bar in the middle. To identify the main colours in each circle, my algorithm first transforms the colour space of the pixels within the circle from RGB to CIELAB. Since CIELAB uses perceptual brightness L^* as one of the values, my algorithm ignores this field and instead focus on the green-red axis a^* and the blue-yellow axis b^* . This is an advantage since it enables my algorithm to identify the colours in the no entry signs, despite the lighting levels. My algorithm then performs K-means clustering on the a^* and b^* axes, to identify the two primary colours in each circle. Since CIELAB is designed to be perceptually uniform, my algorithm then calculated the Euclidean distance from pure white and pure red to the two cluster centers, to identify whether the primary colours in the circle were red and white. If the two primary colours where red and white, and there was more red in the circle than white, my algorithm then performs the line Hough transform on the circle. If the circle contains two approximately horizontal parallel lines, the circle is accepted as another positive no entry sign.

4.2 Visualise

4.3 Evaluate

Table 4: TPRs and F_1 scores of the improved implementation and the differences compared to the previous implementation

Image	Result		Difference	
	TPR	F_1 Score	TPR	F_1 Score
NoEntry0.jpg	1.00	1.00	± 0.00	± 0.00
NoEntry1.jpg	1.00	1.00	± 0.00	± 0.00
NoEntry2.jpg	1.00	1.00	± 0.00	± 0.00
NoEntry3.jpg	1.00	1.00	± 0.00	± 0.00
NoEntry4.jpg	1.00	1.00	± 0.00	± 0.00
NoEntry5.jpg	0.20	0.31	± 0.00	± 0.00
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	± 0.00	± 0.00
NoEntry9.jpg	1.00	1.00	± 0.00	± 0.00
NoEntry10.jpg	1.00	1.00	± 0.00	± 0.00
NoEntry11.jpg	1.00	1.00	+0.50	+0.33
NoEntry12.jpg	0.38	0.55	± 0.00	± 0.00
NoEntry13.jpg	1.00	1.00	+1.00	+1.00
NoEntry14.jpg	1.00	1.00	± 0.00	± 0.00
NoEntry15.jpg	1.00	1.00	± 0.00	± 0.00
All images	0.58	0.73	+0.08	+0.07

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/>.
- [2] *Cascade Classifier Training.* URL: https://docs.opencv.org/3.4/dc/d88/tutorial_traincascade.html.