# Subtask 1: The No-Entry Sign Detector

## Training Performance

Training performance improved rapidly over the three stages. The initial stage had only 100% rate of true and false positives by evaluating every image to be positive. The two next stages quickly reduced the false positive rate to 1.65% and 0.05% respectively. The true positive rate remained at a constant 100% throughout, exposing the model’s bias towards positive images on the training data. This is because the positive images are very straightforward and are tested on the same data that it is trained on.

## Testing Performance

A picture containing text, tree, outdoor, sign

Description automatically generatedCompared to the training, the model performed considerably worse on testing data than training data. The true positive rate averaged 69.3% and an F1 score of 0.516. Due to the testing positives being partially obscured (Figure 2) while varying much in shape and proportion (Figure 3), there is a considerable drop of the true positive rate from the training performance.

Figure : True Positive Rate and False Positive Rate for each stage

The model is unable to spot signs without the high contrast borders that are present in the positive training images and is also unable to distinguish between brighter areas in images and the main white stripe on the stop sign. This is most likely due to the regularity and lack of variety of the positive training images or the model being underfitted.

Adjusting the model’s ‘minNeighbours’ attribute resulted in a worse true positive rate and overall F1 score, however much increased the model’s reliability (Figure 5). This could be argued to perform better depending on the context the model is used for.

Figure : Obscured signs

A picture containing text, sign

Description automatically generatedPerformance could be improved with a wider variety of training images, both positive and negative. A further stage in the model’s creation process could theoretically improve performance by reducing underfitting but experimentation with 3 stages resulted in a massively reduced true positive rate and therefore overfitting.

Figure 5: Min neighbours against various scores

Figure : Mistaken white stripes

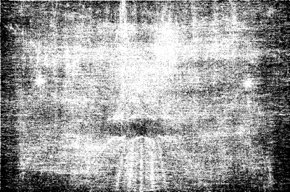
Figure 3: Obscured and difficult orientations

# Diagram Description automatically generatedSubtask 2: Integration with Shape Detectors

A picture containing text

Description automatically generatedA picture containing text, street, sign

Description automatically generatedA picture containing text, building, altar

Description automatically generatedA picture containing text, tree, outdoor, sign

Description automatically generated

A close-up of a person's chest

Description automatically generated with low confidence

A picture containing night sky

Description automatically generatedA picture containing tree, outdoor, night

Description automatically generated

Figure 7: NoEntry12

Figure 6: NoEntry8

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Shape Filtering | |  |  |  |  |  |  |
| Name | TP | FP | FN | TPR | F1 | TP Difference | FP Difference |
| NoEntry0 | 0 | 0 | 2 | 0 | 0 | -2 | -5 |
| NoEntry1 | 0 | 0 | 1 | 0 | 0 | -1 | -1 |
| NoEntry2 | 1 | 0 | 0 | 1 | 1 | 0 | -1 |
| NoEntry3 | 2 | 0 | 0 | 1 | 1 | 0 | -5 |
| NoEntry4 | 2 | 0 | 0 | 1 | 1 | 0 | -2 |
| NoEntry5 | 1 | 0 | 9 | 0.1 | 0.1818 | 0 | -3 |
| NoEntry6 | 3 | 0 | 1 | 0.75 | 0.8571 | 0 | -1 |
| NoEntry7 | 0 | 0 | 1 | 0 | 0 | 0 | -1 |
| NoEntry8 | 3 | 0 | 3 | 0.5 | 0.6667 | 0 | 0 |
| NoEntry9 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| NoEntry10 | 2 | 0 | 1 | 0.6667 | 0.8 | 0 | -3 |
| NoEntry11 | 0 | 0 | 2 | 0 | 0 | -1 | -2 |
| NoEntry12 | 3 | 0 | 4 | 0.429 | 0.6 | -1 | -2 |
| NoEntry13 | 0 | 0 | 1 | 0 | 0 | 0 | -2 |
| NoEntry14 | 1 | 0 | 0 | 1 | 1 | 0 | -2 |
| NoEntry15 | 2 | 0 | 0 | 1 | 1 | 0 | -1 |
| Average | 1.3125 | 0 | 1.5625 | 0.52785625 | 0.5691 | -0.3125 | -1.9375 |

* Viola-Jones detection drastically lowered false positive rate, leaving no false positives.
* Detection reduced the true positive rate significantly however this is fixed by further filtering (see Section 3)
* This however has not contributed to high number of false negatives as only positives have filtered out.
* This detector was designed to be reliable over accurate, however various parameters can be adjusted to make the detector more lenient and give a higher true positive rate.

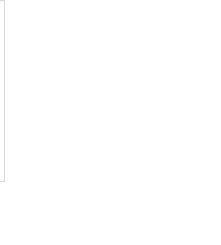
To avoid introducing more false positives into the initial detector, a filtering approach was taken to the shape detector. Detections were either accepted or discarding depending on the model’s certainty of the detection being in the shape of a stop sign. This is more performant as Hough spaces only need to be calculated for an individual detection rather than an entire image while being flexible as the acceptance threshold or importance of individual elements can be easily adjusted.

## Subtask 3: Further improvements

### Parameter Analysis

Each element of the above shape filtering created a probability based on the input value from the Hough space, a minimum, maximum, range of uncertainty and importance.

To optimise these parameters, I generated these values for true detections and false detections.

The best use was for the colour analysis parameters (See below), it is shown in the graph above that a large proportion of true values are seemingly normally distributed with a mean of around 0.42 translating to around half the pixels in the detection being red. On the other hand, most false positive detections have almost no red in them as shown by the false red graph. Therefore, for this parameter, I opted for a minimum value of 0.3, a maximum of 0.6 and a falloff of 0.2.

### Colour analysis

A group of people outside a building

Description automatically generated with low confidenceA picture containing application

Description automatically generatedColour detection was used to further filter the detections. The same probabilistic approach and analytics was used above, except the value was the proportion of pixels in the detection which were above a red or white threshold. First the pixel was normalised by dividing by the median brightness. A pixel is red if the difference between the red component and the others is above a certain threshold (COMS30068 Week 3 Lecture 09 on segmentation). A pixel is white if its brightness is above a threshold and its saturation is below a threshold. Brightness is measured by the sum of channels and saturation is measured by the variance of a channel.

Figure : White detection (left) and Red detection (right)

### Segmentation

Using the colour analysis and thresholds from above, new detections were found to increase the true positive rate. This was first done by partitioning the space into smaller areas with enough red that could contain a detection (shown by blue boxes in Figure \_). This is done by recursively subdividing the space and calculating the proportion of the area with red pixels. Each area is then searched for a detection of best fit. Partitioning allows for less duplicates from crossover detections and faster program execution as less image area needs to be fully searched. These detections are appended onto the previous model. If there are duplicates, they are removed by using the earlier intersection over union calculation. The resulting detections are the again filtered with the above method but with stricter parameters to account for increased false positives. I believe with more time a better F1 score could be achieved either with a more elaborate filter or just more tweaks in the parameters