Autonomous Vehicles

Object Detection

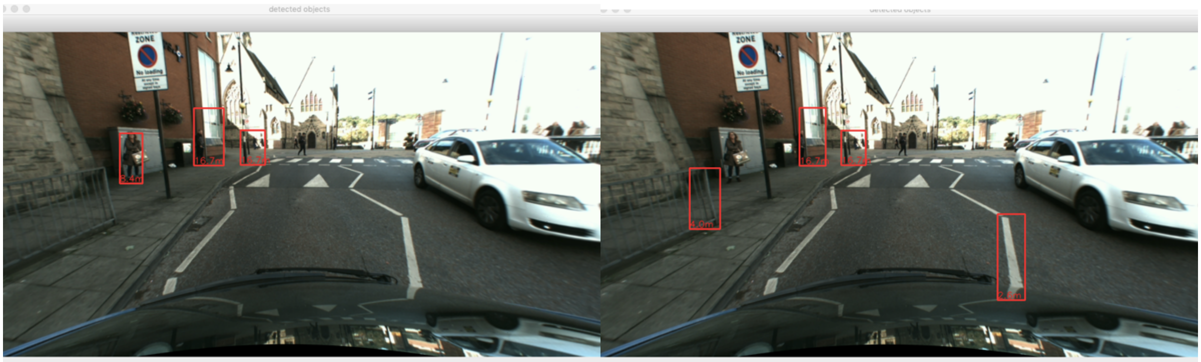
Starting with the code base provided, HOG and BOW were tested by training and testing SVMs. HOGs performance was found to be the superior method of object detection after testing, as the results show.

HOG Test Results:	BOW Test Results:
4.35% error	*30.16% error*
95.65% examples correct	*69.84% examples correct*

Pre-Processing

Brightness and contrast were altered images were fed into the HOG detector. When increasing brightness, the recognition performed poorly, with more false positives and less successful identifications. Altering contrast proved effective, with tests at different level showing that the best results occurred at contrast +70.

Contrast 70 | Contrast 0



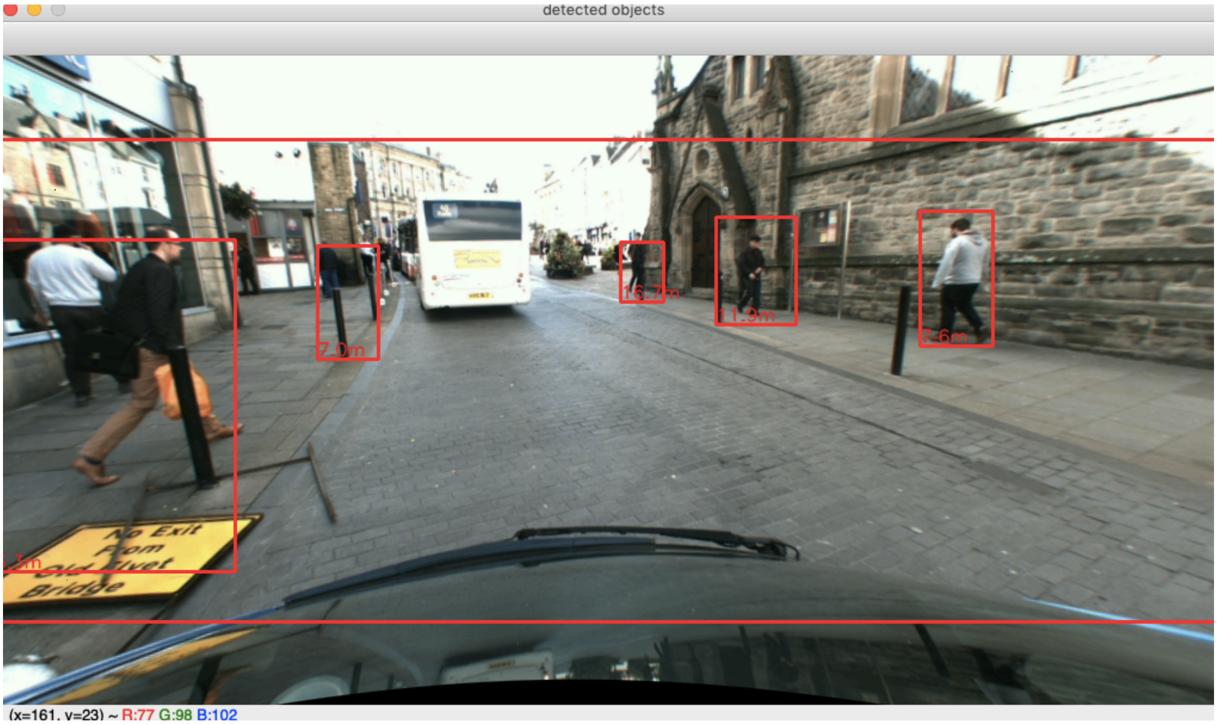
Higher contrast detects 3 pedestrians correctly, while normal contrast detects 1 correctly and makes 3 false detections

Canny and Sobel Edge detection were implemented to improve the rate of detection, however both methods resulted in a reduced performance. HOG trained to edge-based images produced a large number of false positives, associating various shapes with pedestrians.

Heuristics to Increase Efficiency

Region of Interest

In order to improve performance and efficiency, a region of interest was employed. This filtered out objects identified by the selective search if they lay above or below thresholds of 70 pixels, meaning that areas of the sky and car bonnet were dismissed whilst prioritising the road of pavement. As well as increasing efficiency, this method reduced false positives where pedestrians were unlikely, thus improving performance.



Illustrated are the boundaries of the region of interest

Filtering by Shape

The software also filters all objects failing to meet criteria which make them likely humans: width at most 75% height, minimum size, height at most 5x width. This reduced the number of false identifications, and saved time processing objects which could not be human.

Depth Calculation

To calculate depth, the disparity image was converted into a dictionary where points for which there is disparity are mapped to a corresponding Z value – the distance to the car.

To calculate the distance to an object, because pedestrians stand in front of the background, a central box is created within the object rectangle. The distance to the object is then set to the lowest pixel distance in the box. This is more efficient than considering the entire region, and avoids including the foreground, as well as being safer than a mean average.

Depth Filtering

To aid in pedestrian identification, while calculating closest distance, the average is also taken and compared to pixel boxes in the top corners of the object region. If the central area is on average 25cm closer, the object passes, otherwise the object is assumed not human. This was an effective method for removing false positives, as humans are central in the region while flat surfaces were ruled out.

The left side of the left image has no disparity, so for regions with x coordinates < 50px a depth is not required but pedestrians can still be identified. They are automatically assigned a distance of 1m, as they are assumed to be on the pavement. Objects calculated to be further than 30m are also dismissed, reducing the number of small objects identified far away. The identification performed poorly at long distances where objects are small, and so this method reduced false positives. Furthermore, pedestrians further than 30m are not of immediate concern and so not identifying them is low-risk.

The Disparity image demonstrates how pedestrians stand out from the background (The image also shows the blind spot at the left)

Selective Search and Sliding Window

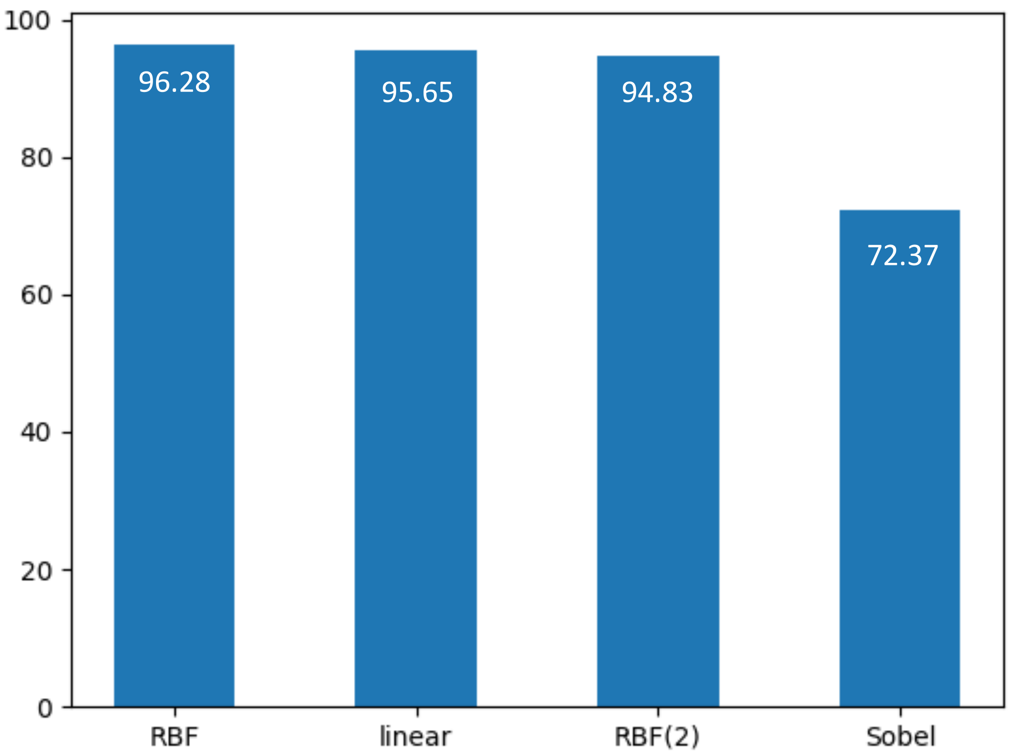
Sliding window was originally the method employed to search the image, however because of its exhaustive nature it was inefficient and produced a large number of false positives.

Deploying selective search proved far more effective, as it focussed analysis on only areas of objects, saving time running identification on background, and improving the chances of finding pedestrians.

Training SVMs with Different Kernels, Filters and Parameters

Having initially used a Linear kernel experimentation was done on whether different kernels produced better results. The RBF kernel performed best, improving on linear, and is used in the demonstration software. Another version of RBF was also tested, using larger sample sizes for Data window offset (5), negative sample count (15) and positive count (10) as opposed to the parameters used before (5, 15, 10), however it proved marginally less effective.

The parameter for number of iterations was also reduced to 100 from 500 when implementing RBF. This decreased training time, and contributed to the improvement in overall performance.



The Graph shows the performance in testing of different SVMs, with RBF the most successful (96.28%)

Performance

The refined HOG detector performed well on the input data, prioritising finding most pedestrians in a given scene while resulting with few false detections. In scenes without any pedestrians, often no objects were falsely detected. This is a great improvement on the first trial, which discovered some pedestrians but resulted in many false detections.