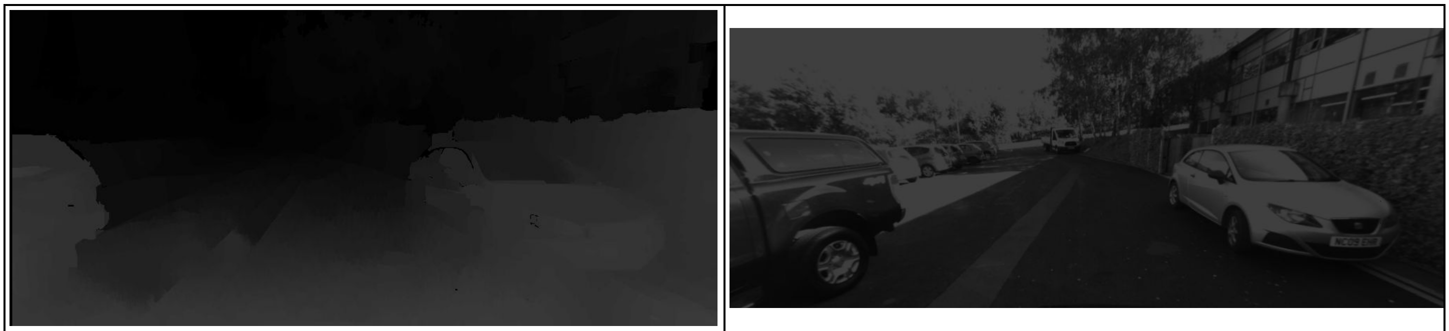
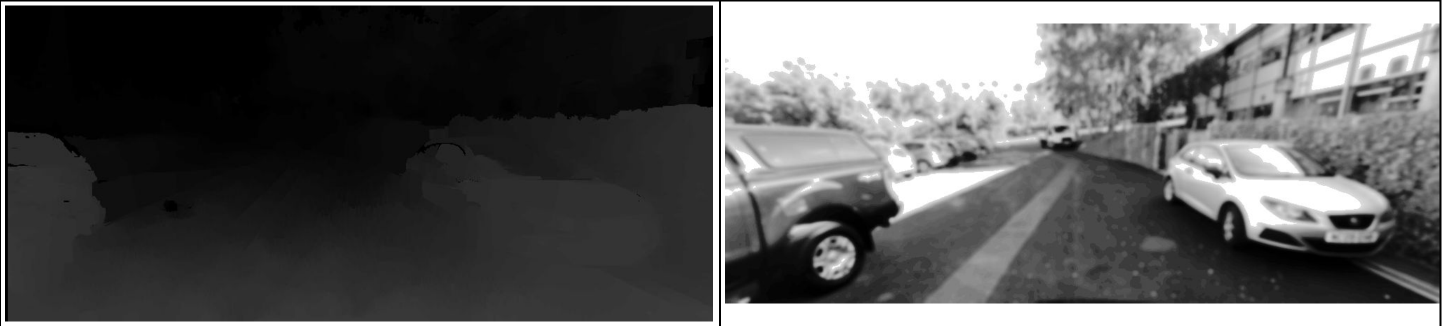
L3 Computer Vision - Jun Eeo (njzj22)

1. Preprocessing

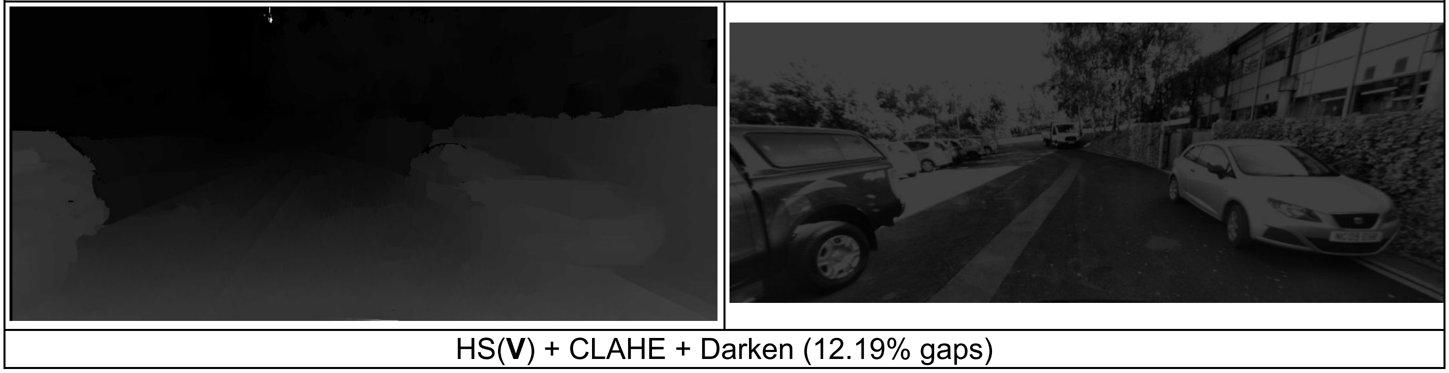
We first convert the image into grayscale by taking the maximum of each RGB component (equivalent to the V calculation in HSV), performing CLAHE on V, and raising it to power 0.75. One alternative approach taken was to apply the bilateral filter followed by histogram equalisation. We found that HSV + CLAHE is more performant and gives better results (subjectively less noise with fewer gaps in the disparity map across a set of test images).



Grayscale + darkening (13.14% gaps)



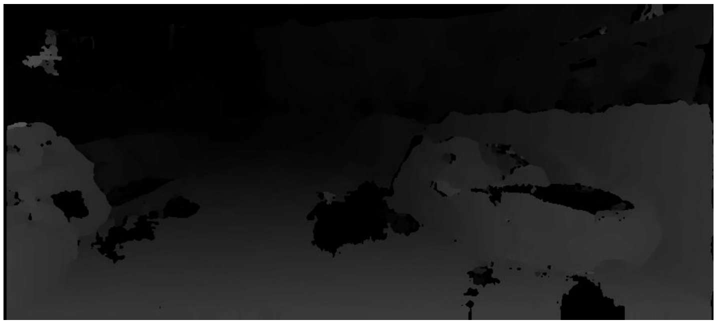
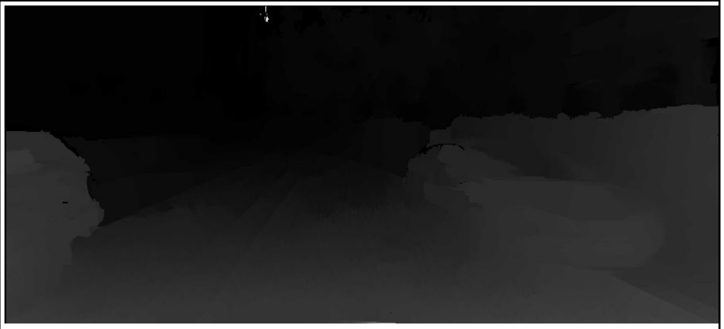
Bilateral Filter + Grayscale + Histogram Equalisation (12.72% gaps)



HS(**V**) + CLAHE + Darken (12.19% gaps)

2. Disparity Map Calculation

For computing the disparity map, we used the Stereo SGBM algorithm (with 128 max disparities) alongside the WLS Filter with lambda = 8000 and sigma = 1.2. This approach yields a much higher quality disparity map with much less gaps than purely using the SGBM algorithm.

	
SGBM + Speckle Filter	SGBM (Left and Right) + WLS

3. Distance Calculation

Given a bounding box of the match, we estimate the distance as follows. We compute the distances by first computing an Otsu threshold on the histogram of disparity values, the aim being to find a separation between foreground and background values. The estimated disparity is taken as the maximum of the mode of the disparities and the median of the disparity values above the threshold. We take the maximum under the assumption that it is always better to under-predict the distance rather than over-predict.

In Figure 1 we show the histogram of the disparity maps for the match windows. In most cases the estimated distances are reasonable, except for the 3.98m car where the predicted distance was much less than the actual distance. The reason we overestimate the disparity in this case is because the bounding box overlaps with the car that is 3.48m away.

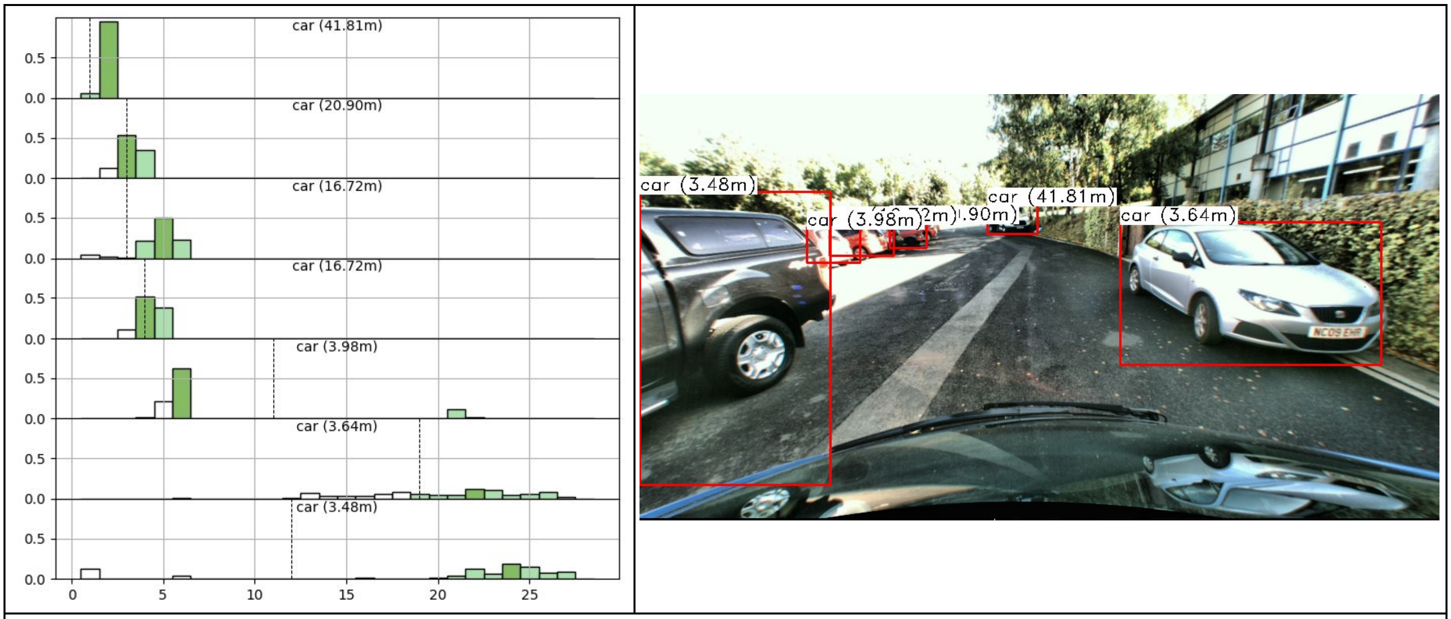


Figure 1: Histogram of disparities for the corresponding image. Light green = above threshold, and green = mode. The dotted line shows the Otsu threshold.

Figure 2 shows in some cases, our estimator corrects for the faults of purely using either the median or mode (for the person). In most cases however, the mode is the “right” disparity estimate.

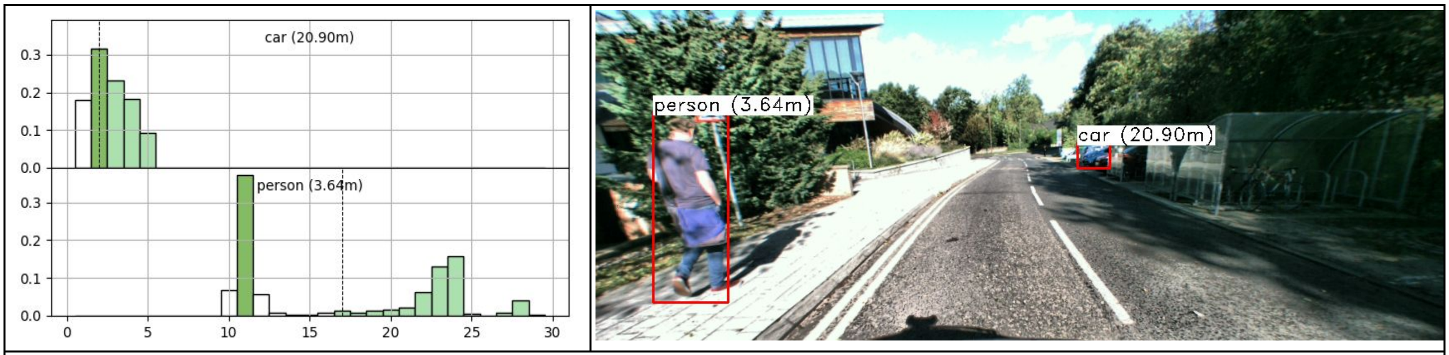


Figure 2: Using purely the median/mode would have overestimated distance.

Our distance estimator is stable for objects with good disparity information, as shown in Figure 3 - observe that the stationary bus has the same distance estimate for the next few frames (starting from 1506942718.476805).

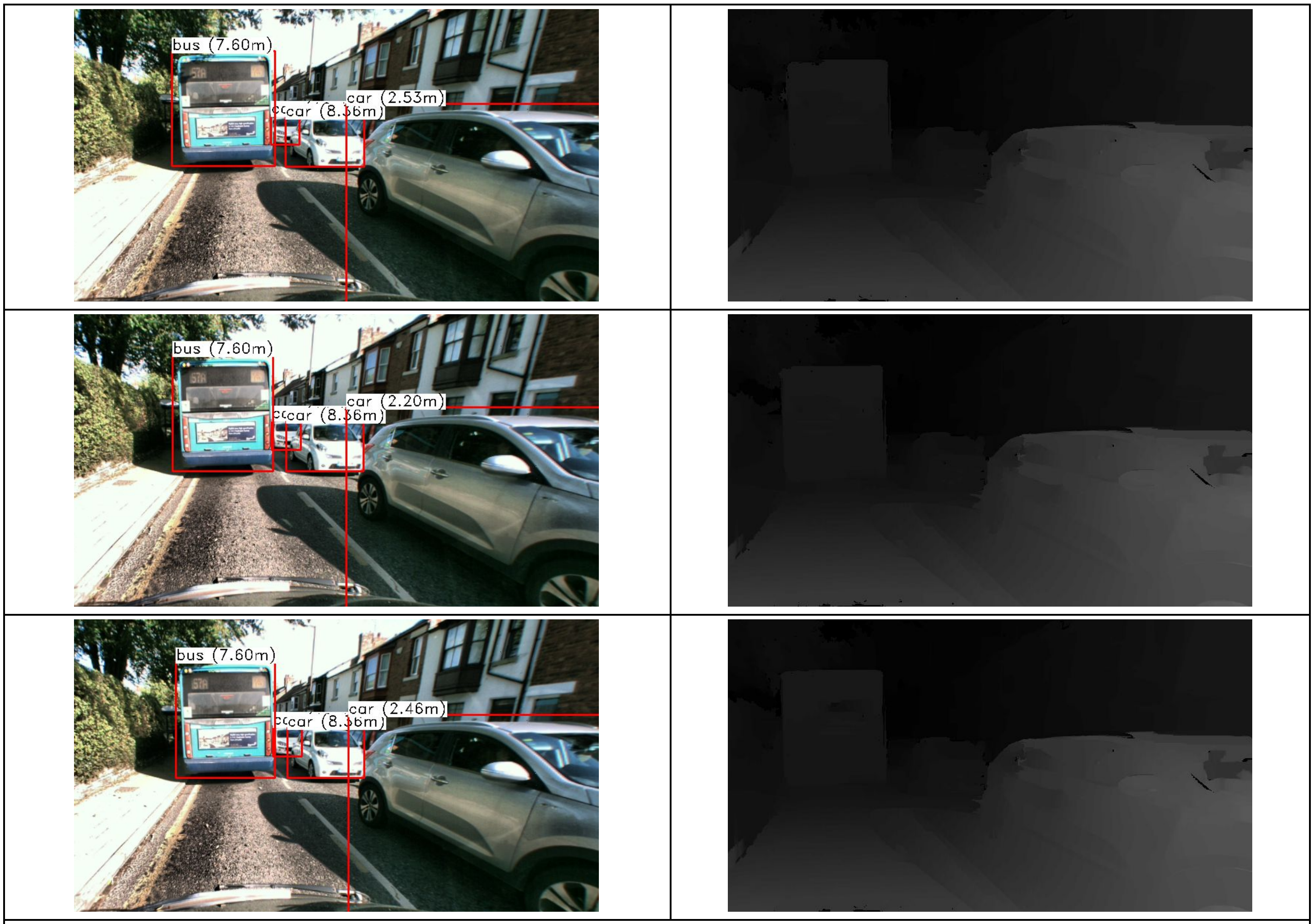
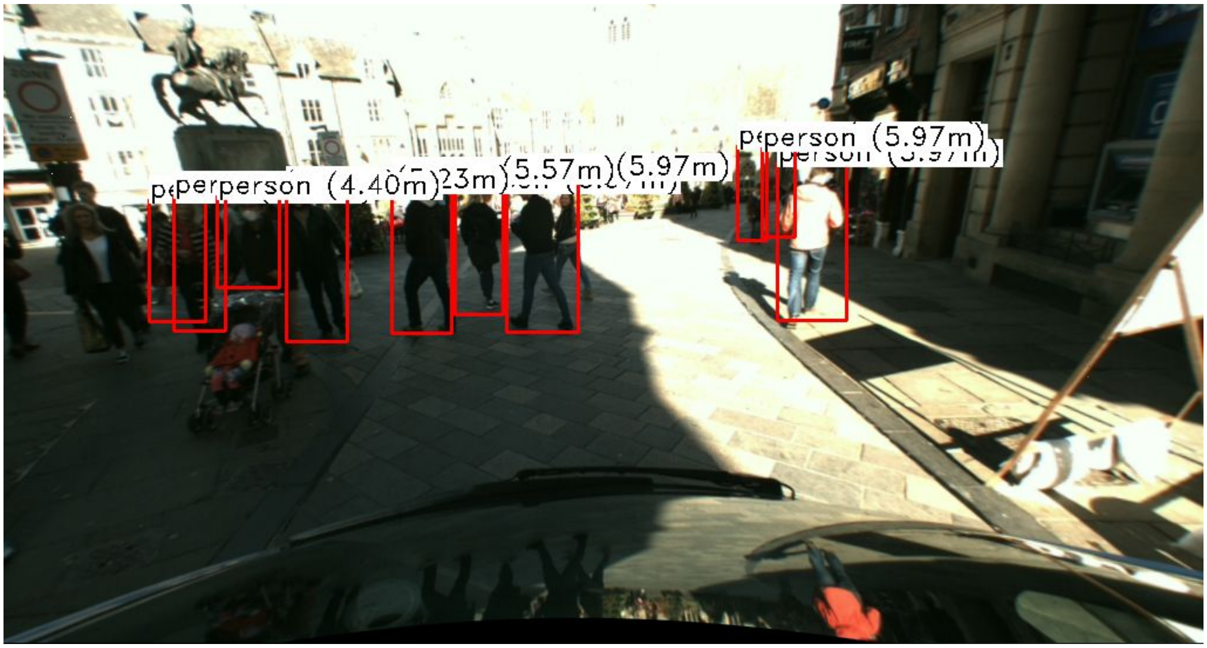


Figure 3: Distance estimate for the car on the right fluctuates slightly, due to the periodic noise in the left and right image because of the shadows cast by the moving tree branches.

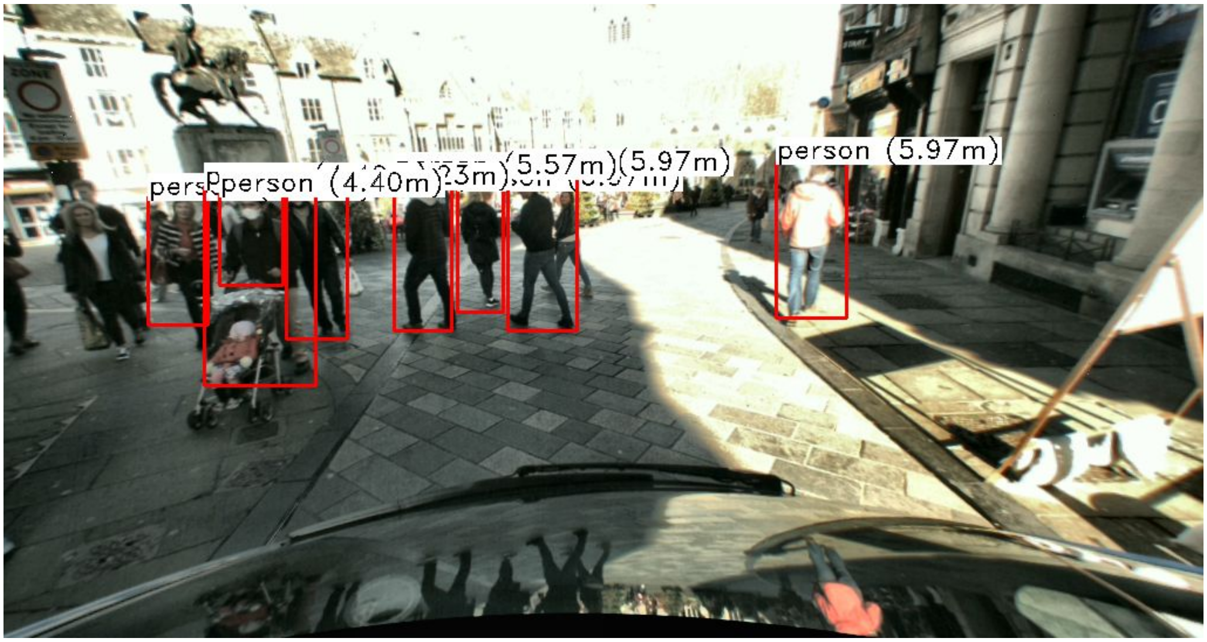
There were attempts at improving disparity estimation by, e.g. taking the median of the side where the mode lies, but we found that this wasn’t good for objects close up to the car.

4. Preprocessing for Object Recognition

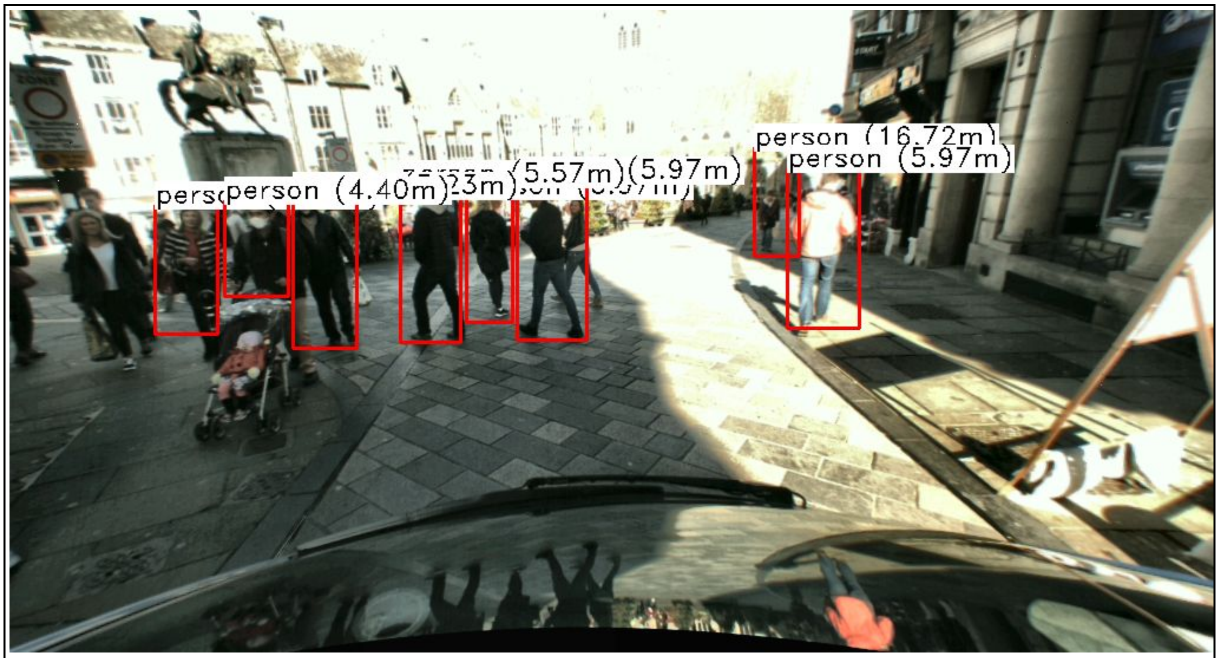
To improve object recognition performance by YOLOv3 we performed CLAHE on the V channel of the HSV representation of the image. Below we show the comparison between equalisation on different colour spaces - we find that HSV gives us better bounding boxes and fewer false positives.



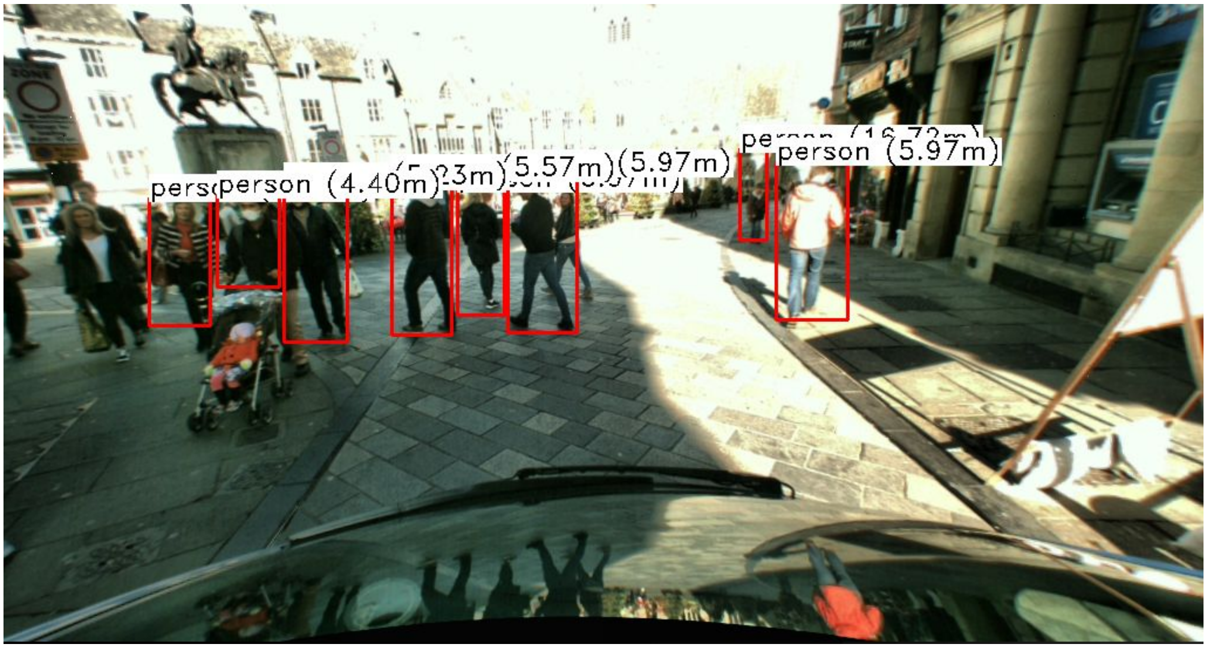
No equalisation - note false positive (person detected twice) due to poor contrast.



(Y)UV equalisation - missed the person that is further away.



(L)AB equalisation - better than YUV.



HS(V) equalisation - better bounding boxes overall without false positives.

Sparse Stereo Comparison

For sparse stereo vision, we largely used the same preprocessing - CLAHE on V-channel to normalise the two left and right images before extracting ORB feature points. Distance was estimated using just the median - we assume that we have good disparity information.

Figure 4 shows that most of the distances in the dense stereo method agree with sparse stereo. However, sparse stereo has some trouble dealing with objects that are closer - for instance in the first image, the left car is marked as 3.22m with dense stereo but 28m with sparse stereo (same for the truck in row 2). Sparse stereo tends to deal with occluded and small/far-away objects better - see the first and last rows of Figure 4.

One issue with sparse stereo is the potential lack of (relevant) feature points, especially in scenes with trees, which in some cases causes detected objects to not have disparity estimates (e.g. Fig 5).

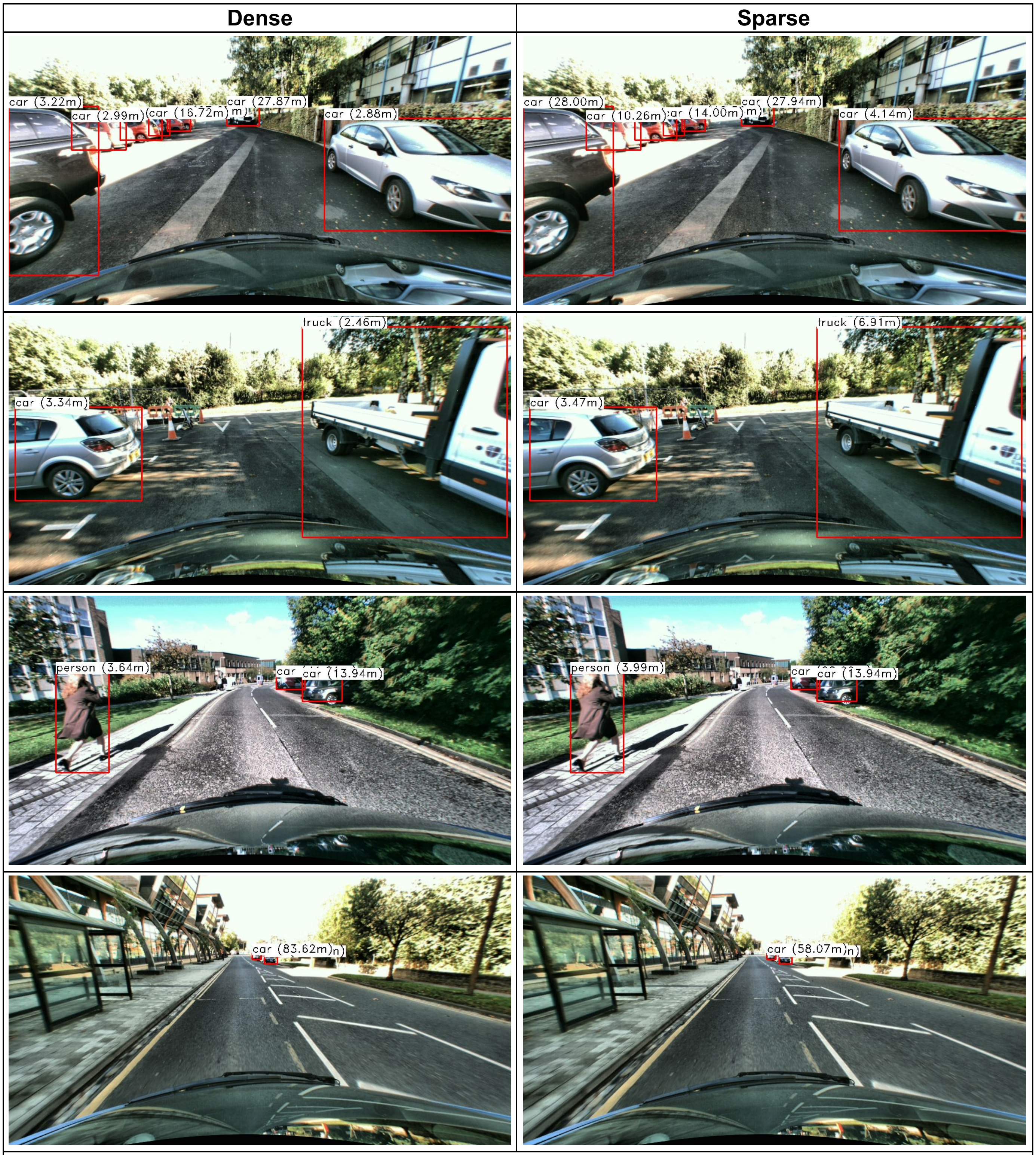


Figure 4: Comparison between distance estimates with sparse and dense stereo.

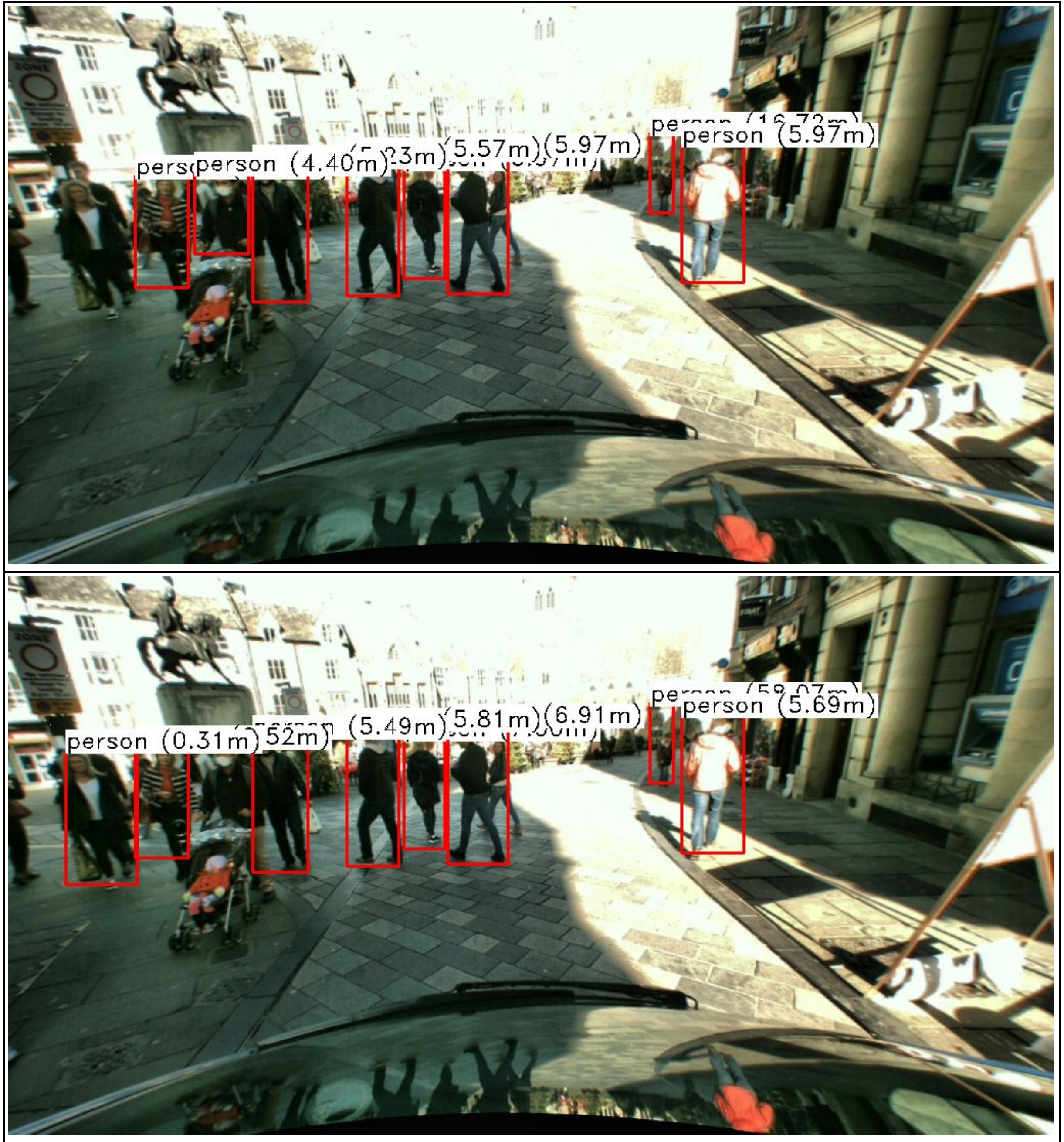


Figure 5: Dense (top) vs sparse stereo (bottom) for scenes with many objects.