Computer Vision Report

Creation process

Before any experimentation could occur, a baseline system needed to be created. I began by stripping the given yolo.py to its main functions and modularising it for use on individual frames. I then used parts of the given stereo_disparity.py to develop two functions for dense disparity distance calculation. First, yolo would detect the objects of interest and give their regions. Next, it would take the left and right images and construct a dense disparity map. Finally, it would take different regions of the objects and work out an average disparity value to then work out a distance using the focal length and camera baseline of the stereo capture device. Once I had this basic system working, I could begin to experiment with different techniques of optimization.

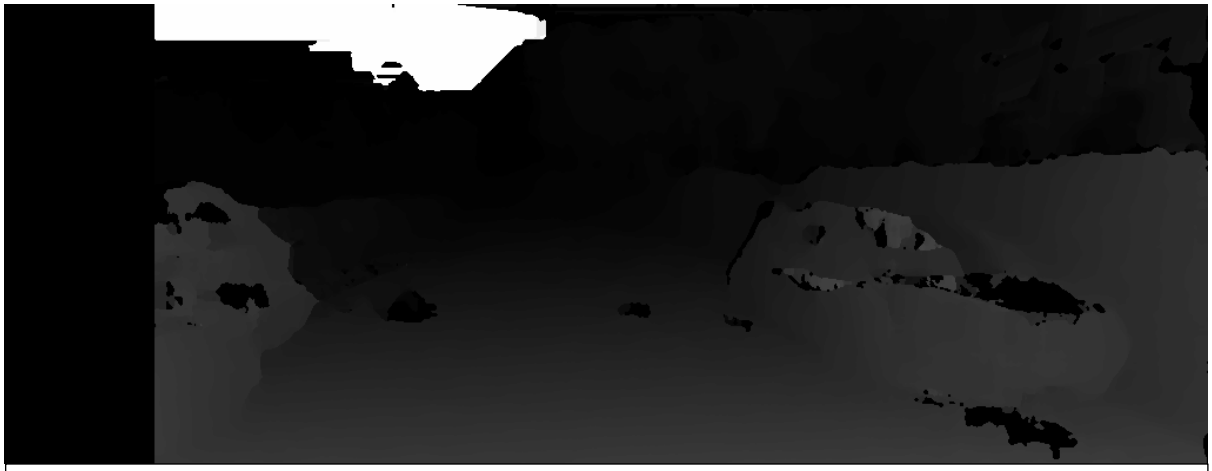
Making distance calculations better

The first changes I made were with the method of calculating disparity average. Initially I just used the centre pixel of the feature region to work out distance, this began to fail when we had overlapping objects as the centre pixel would be of an object in front. I then used a standard mean of all values, however this proved to be ineffective as the feature region was a box not a perfect outline of the feature meaning there were many zero values skewing the mean causing distance values to be much larger than they should have been. To compensate for this, I tried using the median value (ignoring any NaNs) and attempted to use a nonzero mean, the effects of which can be seen in set1 of the appendix. I also tried to implement a method that uses percentiles however I couldn't successfully achieve this. In theory, a percentile method would have ignored outliers and thus avoided being skewed by noise (which happened with the mean).

Pre-processing

The next thing I noticed was that many of the images had a low contrast. To remedy this, I used a form of histogram equalization called Contrast Limited Adaptive Histogram Equalization (CLAHE) which works by taking small regions (tiles) and applying equalization on those, rather than the entire image. This proved to work and you can see the effect of the filter in set2 of the appendix. Whilst the filter didn't seem to have much effect on distance values, it had some success in helping yolo detect objects in poor light conditions.

The next experiment was to apply a filter to the disparity map. The filter I tried was the Weighted Least Squares (WLS) filter. The WLS filter smooths the disparity map and makes it more uniform. This seems like it should help with the distance calculation though, in practice, not much change was seen and in some cases the filter made things worse (example on next page). This could be because smoothing causes the image to lose detail and thus lose valuable information that could have helped with distancing. I also apply a noise filter to the disparity map to lower the amount of noise as this would help to provide a better distance average. The next page shows a WLS disparity vs regular disparity.



Normal Disparity map of first frame with basic noise filtering and CLAHE.

WLS Disparity map of the same frame. The smoothing seems to have merged a lot of things together causing a loss of detail and messing with the distance calculations.

Issues

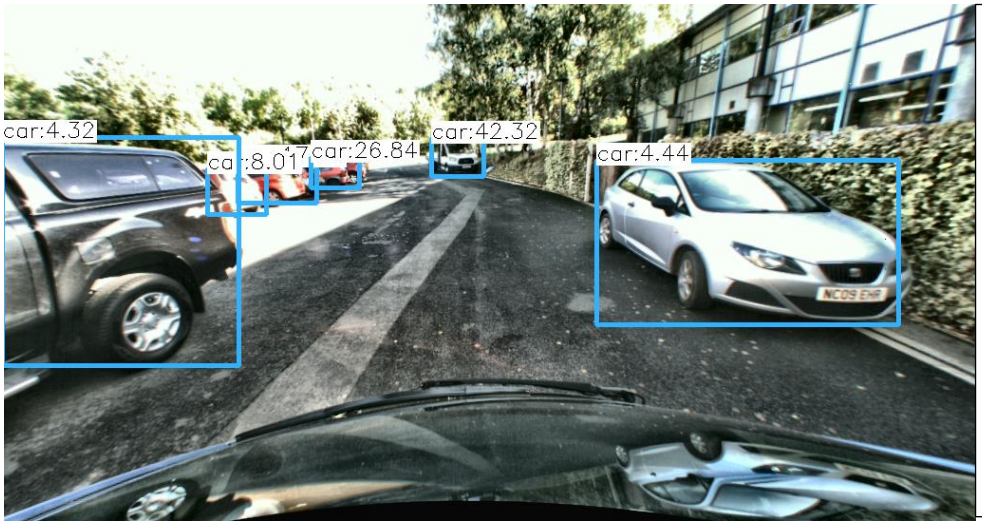
An ongoing struggle with the system was when yolo would detect a very distant object but the disparity map wouldn't have any real information on it causing wildly large or varied distance values that were obviously incorrect. The only solution I could think of is to just ignore the entire object if a suitable distance couldn't be found, however, I thought it would be more beneficial to know an object exists rather than to just ignore it entirely.

Conclusion

In summary, I have experimented with various different techniques throughout this coursework to attempt to increase the robustness of the system. Whilst not all of these have proved effective, they have all lead to a solution that is a suitable prototype for object distance detection. YOLO has been able to find most of the objects in the scene (helped a little by CLAHE) and the disparity maps seem to have given enough information to get a reasonable distance estimate.

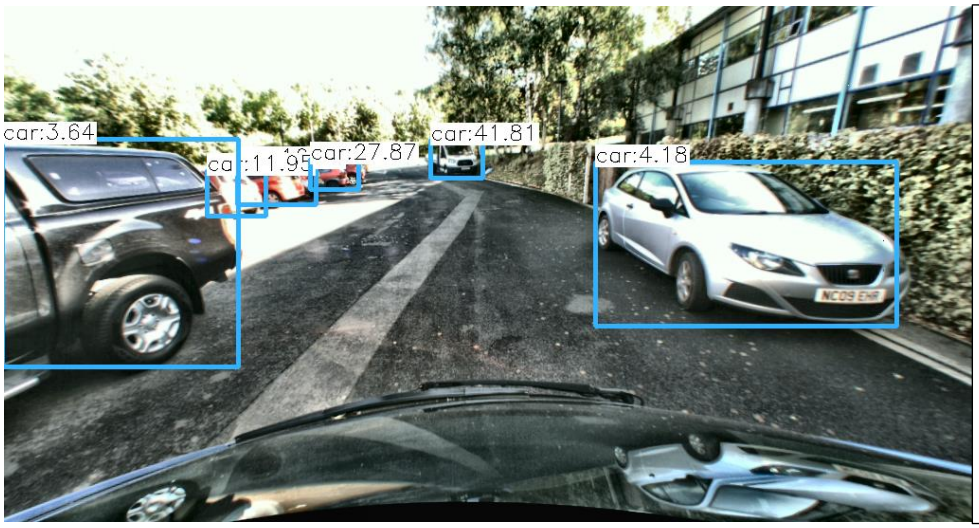
Appendix

Set1 - disparity averages



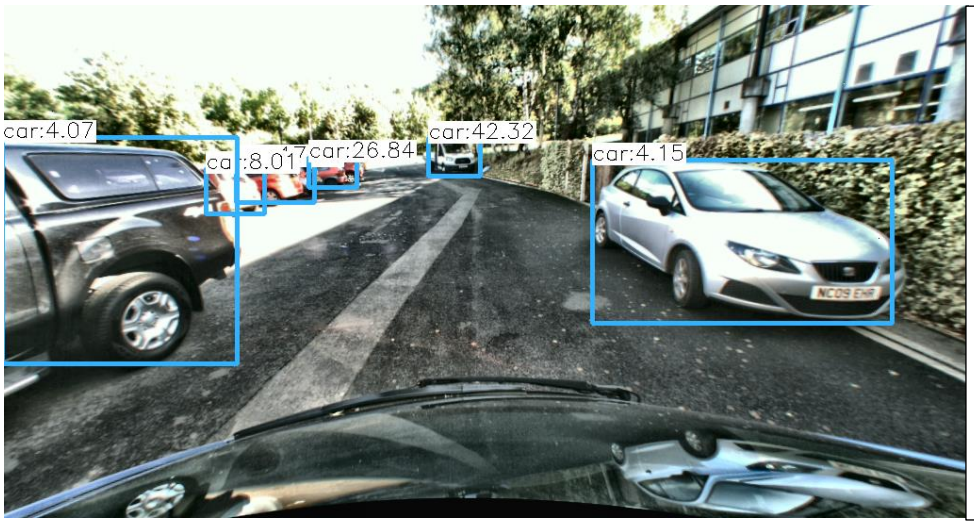
Mean values.

Compared to the other techniques the distances have been overestimated



Median Values

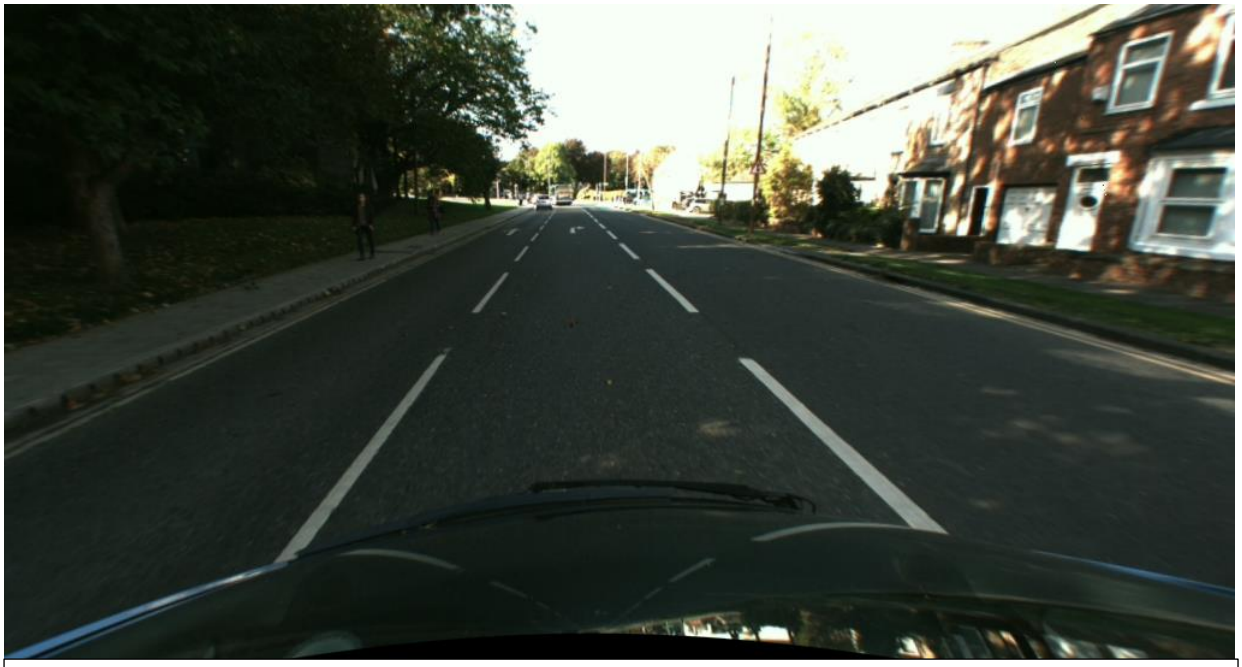
Better results than the mean. Car:11.95 seems to be an anomaly, maybe caused by occlusion?



Nonzero mean

Seems to be the average of the other two methods. Only issue with this method is the addition of NaN values for distances.

Set2- with and without CLAHE



Low contrast image taken by stereo camera. People on the left is very hard to see.

Image equalized with CLAHE, much easier to see the people on the left-hand side