**Report**

**Discussion / detail of solution design and choices made (400)**:

The goal of this project was to develop a system to correctly detect pedestrians and vehicles from the scene in front of you and estimate the distance to these objects. I made use of object detection through YOLO, and implemented two approaches to stereo vision: dense and sparse. The dense approach I consider my main solution, with sparse being an alternative approach to reach the same conclusion.

My Solution – Design Choices

My improvements to the initial integration of dense stereo vision and YOLO:

* **Histogram Equalisation**: Illumination in the provided dataset was an issue, so an attempt to reduce this before calculating the disparity map was made. I used CLAHE to do so – an advanced histogram equalisation method

PHOTOS

* **Stereo\_SGBM parameters**: Adjusting the input parameters for the SBGM object was critical in producing an accurate, but somewhat noiseless, disparity map. Many parameters allowed for the removal of later functions calls – e.g. filterSpeckles was integrated into the object itself through several of its founding arguments.

PHOTOS

* **Calculating distance**: For each object a Gaussian kernel was generated (based on the size of the object detected). The convolution of this kernel with a matrix of disparity information (centre: centre of object’s box) is then performed, and the resulting disparity is used to calculate distance.
* Altering the size of the kernel was imperative to success here as without doing so further objects would have poor distances or the time to calculate would significantly increase.

My changes that proved less effective:

* **Weighted least square (WLS) filter**: The use of the WLS filter along with the use of a left-right-consistency based confidence was attempted to improve the quality of the resulting disparity map (through ‘filling in the holes’). The hope here was a smoother, less noisy disparity map would subsequently improve distance calculations. However, I found that the removal of noise also removed much useful information from the map, with this causing a decrease in distance quality.

PHOTOS

* **yolo edge detecting**: The idea of enhancing the edges of an image before detecting objects seems sound in principle, so an application was attempted. A simple smoothing technique was used with a variety of matrices. Although

\*Research\* Why things weren’t perfect? A main issue came from the hardcoding of the baseline and focal length. These should be taken from the calibration matrix

**Qualitative/Quantitative evidence of performance (150):**

**Comparison between Sparse and Dense Implementations (150)**: