# Baseline Evaluation

The baseline implementation was built using the popular YOLO v5 model architecture, known for its good accuracy and particularly impressive frames per second (fps). Two different models were created, the first of which was made from scratch on the road sign dataset, consisting of 877 images classified into 4 different categories with bounding boxes. The second model was a partially trained model that was then fine-tuned to the 877-image dataset used. The model can be tested using an anaconda environment, with notable required packages being NumPy and Pytorch, as well as some smaller packages such as tqdm.

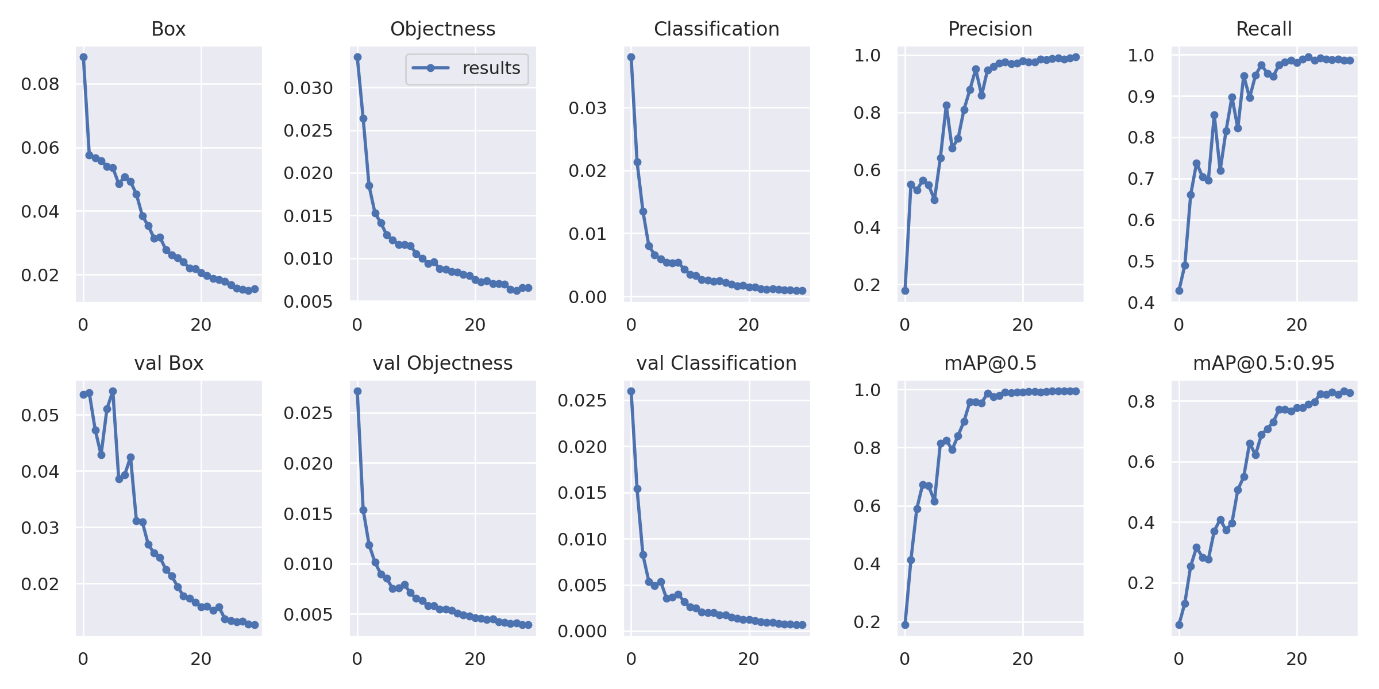
The models were initially going to also be trained using the anaconda environment on Windows 10, however there was issues and incompatibility regarding CUDA and cuDNN and so the popular GNU Linux distro Ubuntu 20.04 Desktop LTS was used instead. This is because CUDA can be installed much faster and easier, as well as less system overhead in general due to the more lightweight Operating System. Progress and comparison of the model is observed using WandB (Weights and Biases), which is an online, remotely accessible tool that gives the console output from the training, as well as graphing and displays on how well the model is performing on the dataset (with images). This allows to very easily compare models, such as the models created as the baseline implementation. This also means that any future models created with different parameters during iterative development can also be compared easily.

Both baseline models were created to an arbitrary count of 30 epochs, with the same resolution of 640x640 and a batch size of 16. These parameters are also very easily changeable in a single text file that can be used to begin training. Any training that has been started can also be resumed simply by changing the beginning weights file to the “last.pt” weights file in the “runs” directory. This means that if a long training session needed to be paused, or there was a system crash, the model is not lost.

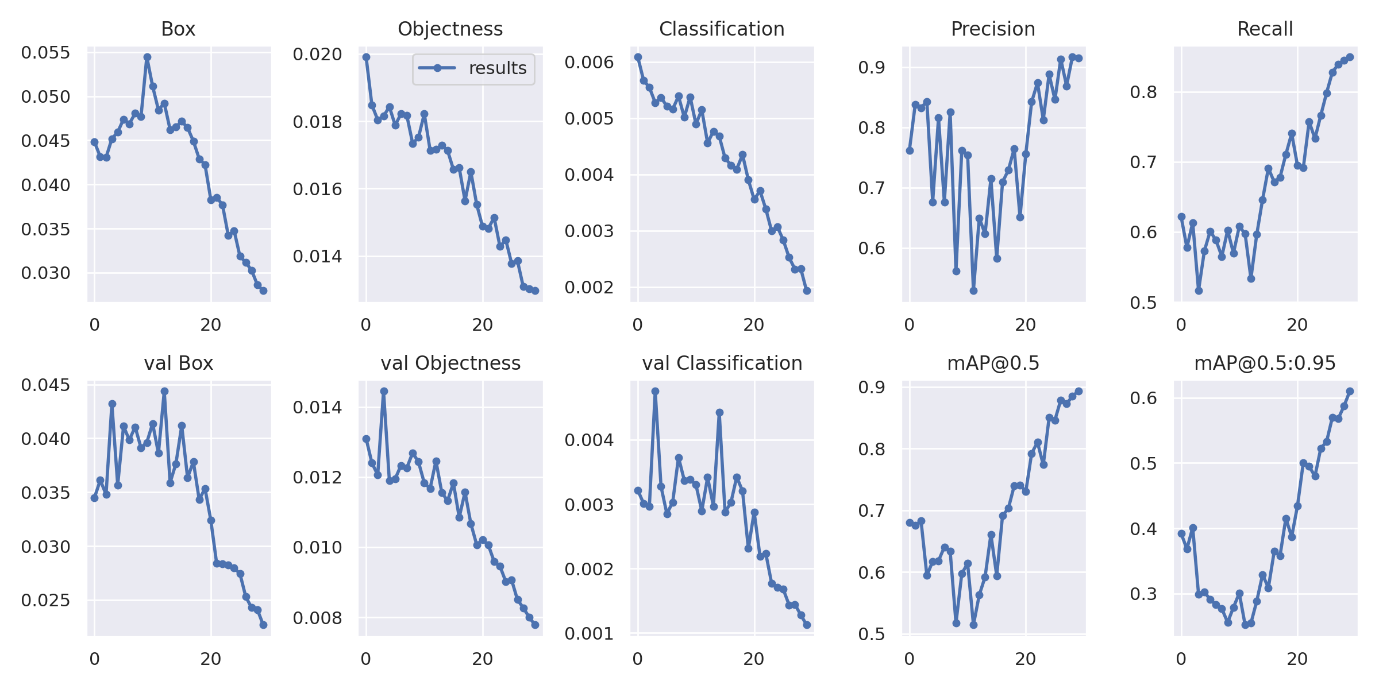
Training is possible to complete without CUDA, cuDNN or a GPU at all, i.e., on Windows, although is very slow without GPU acceleration and requires a lower batch size to avoid memory limits. Testing however, especially for single images or short videos, is not as fast as it would be on a GPU but still very possible without one and therefore can be done on a CPU in a reasonably short amount of time. This means that the testing and execution can all be done from within the anaconda environment, and any training of models done on Ubuntu.

A brief evaluation of both the fine-tuned and from-scratch models show that the fine-tune seems to be far more reliable than the from-scratch design, scoring around 20% higher in previously unseen images. This is also supported by the graphs supplied by WandB, where from-scratch seems much more erratic whereas fine-tuned seems much smoother, suggesting that the from-scratch is likely overfitting.

Fine Tuned:



From Scratch:



All diagrams supplied by wandb.ai