Proposed solution for automating lane detection:

The data provided for automation is RGB image with corresponding pixel-wise mask. Hence the problem statement is based on image segmentation.

The task of image segmentation is to train a neural network to output a pixel-wise mask of the image. This helps in understanding the image at a much lower level.

For precise lane detection on the road I have used the U-Net network. The u-net is convolutional network architecture for fast and precise segmentation of images. Up to now it has outperformed the prior best method (a sliding-window convolutional network).

I have resized the images to 640 x 480 for training the model. We can also consider 480 x 360 but the details in images were more so decided to go for slightly bigger size.

Considering that we have images in a single folder that we want to use for lane prediction, passing the folder to predict.py will give us a predicted mask.

Since U-net training is done from scratch(no transfer learning) it required many epochs to give decent results.

The training results shown below are on 200 epochs with threshold > 0.5:
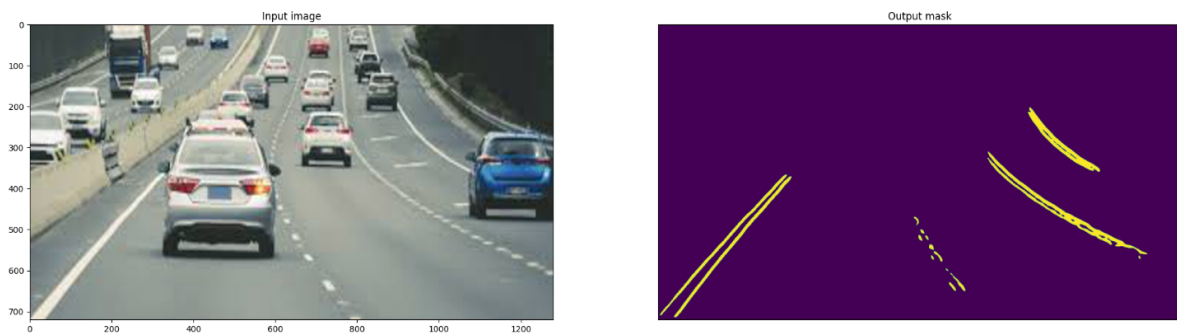
We can see in time-series graph that the loss is further decreasing and can be trained furthermore for better results and accuracy.

INACTIVE

Filter runs (regex)

Filter tags (regex)

Run

-

Pinned

Pin cards for a quick view and comparison

Loss

train