**Assignment Solution**

**by**

**Abhishek Tanksali**

**Assignment Objective:-**

To create a Lane Detector based on a Deep Learning algorithm implementation to automate the lane detection process from street images that can be used by an autonomous vehicle so that it can understand the content in the images and thus make informed decisions.

**Data:-**

Dataset contains 501 JPG images of road lane marking dataset with their masks obtained by drawing the lane marking labels in such a way that everything else in the image, but the lane markings are colored black and the lane markings themselves are colored white.

**Approach:-**

* Input images are RGB images and output label images are mask images. Data is splitted into train (70%), validation(15%) & test (15%).
* Objective is considered as an Instance Segmentation problem.
* Instance segmentation ENet model was tried:-
* ENet:-

(The Research paper for Enet can be found at the link: <https://arxiv.org/abs/1606.02147>)

As per this paper, ENet is up to 18% faster, requires 75% less FLOPs, has 79% less parameters, and provides similar or better accuracy to existing models such as UNet

* ENET Model’s code was borrowed from GitHub, trained on training data, evaluated on validation/test data.

Features included in the workflow:

* Image size: 720p x 1280p
* Train, validation generators are used for code to be memory efficient
* MeanIOU metric is used as an evaluation criteria. It was borrowed from <https://github.com/davidtvs/Keras-LinkNet/blob/master/metrics/miou.py>
* Adam optimiser & binary cross entropy loss function is used
* Due to memory & time limitations and CPU framework, model training is done only for 5 epochs with 500 training steps.
* Learning rate of 0.001 is tried.

Features not included :

* ReduceLRonPlatueau – Learnng rate reducer
* EarlyStopping if loss is not decreasing
* ModelCheckpoint to save best model
* More training for higher number of epochs & higher number of steps
* More training and validation images
* Image data augmentation (zoom/rotation/horizontal flip/vertical flip/rescaling/different image format like PNG/brightness/shear range etc)
* Other models or loss functions if any

Model is under-trained. Mean IOU of 55-56% is achieved which is pretty low.