BDD Lane Detection

Ujjwal Singh Rao

January 2023

Objective

The goal of this exercise is to create a lane detector using deep learning models. As the name suggests - this model should be able to highlight the lanes in a given input image. One such example is shown below.

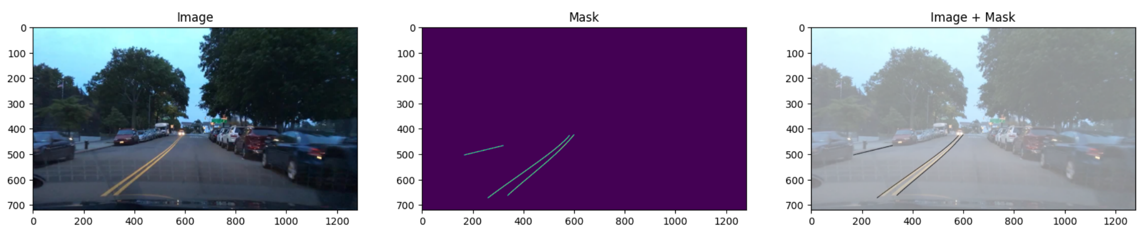

Figure 1: Sample Image

The first image to the left shows the input to the model. The image is the middle is the mask which highlights the pixels in the input image that are of interest. This mask will act as labels in the supervised learning process for training the model. The last image on the right overlays the image and mask for ease of understanding.

Dataset

A small sample of 3.5K images were sourced from the original 100K dataset for this exercise. The number of images in training and validation sets are mentioned in the table below. All the images are of size 720x1280 pixels.

Training	Validation
2976	454

To compensate for the small size of the data set, the below mentioned augmentations are used.

- Random Crop of 720x720 pixels
- Random Horizontal Flip

Figures 2 and 3 are a few examples of how the input images are modified during the model training process using these augmentations.

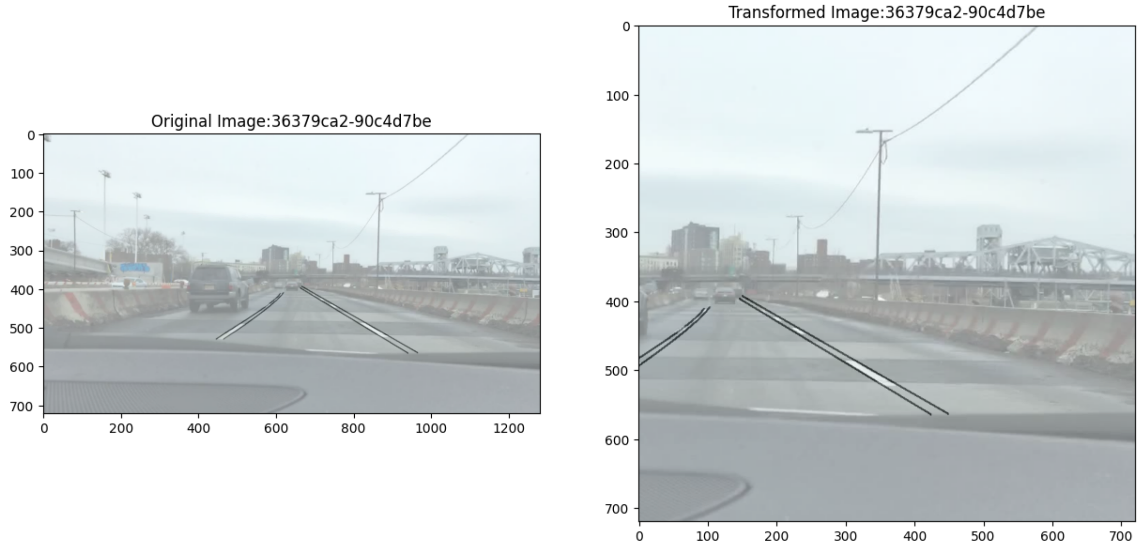

Figure 2: Data Augmentation: Random Crop

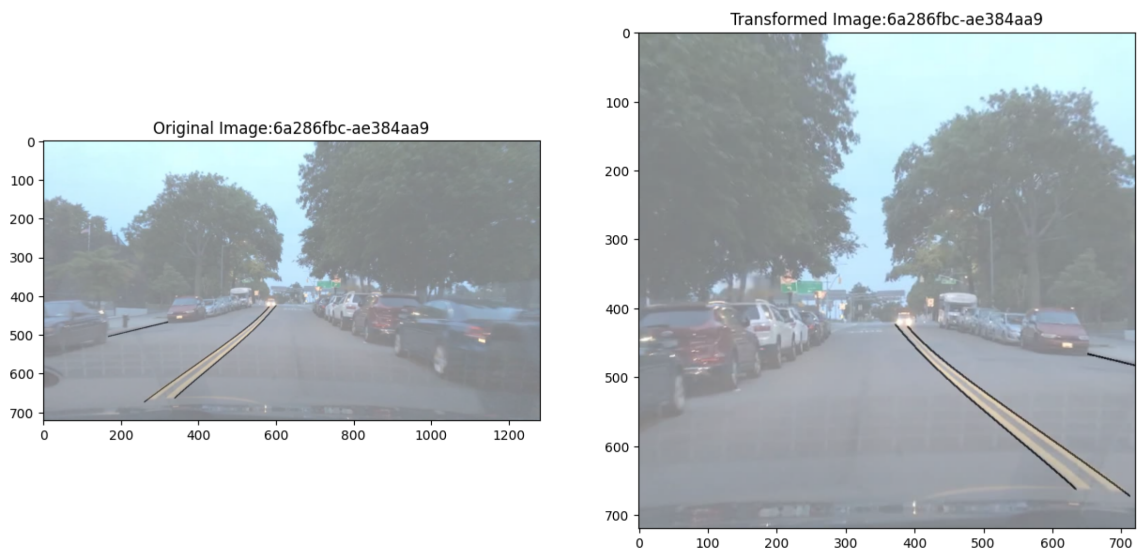

Figure 3: Data Augmentation: Random Crop + HFlip

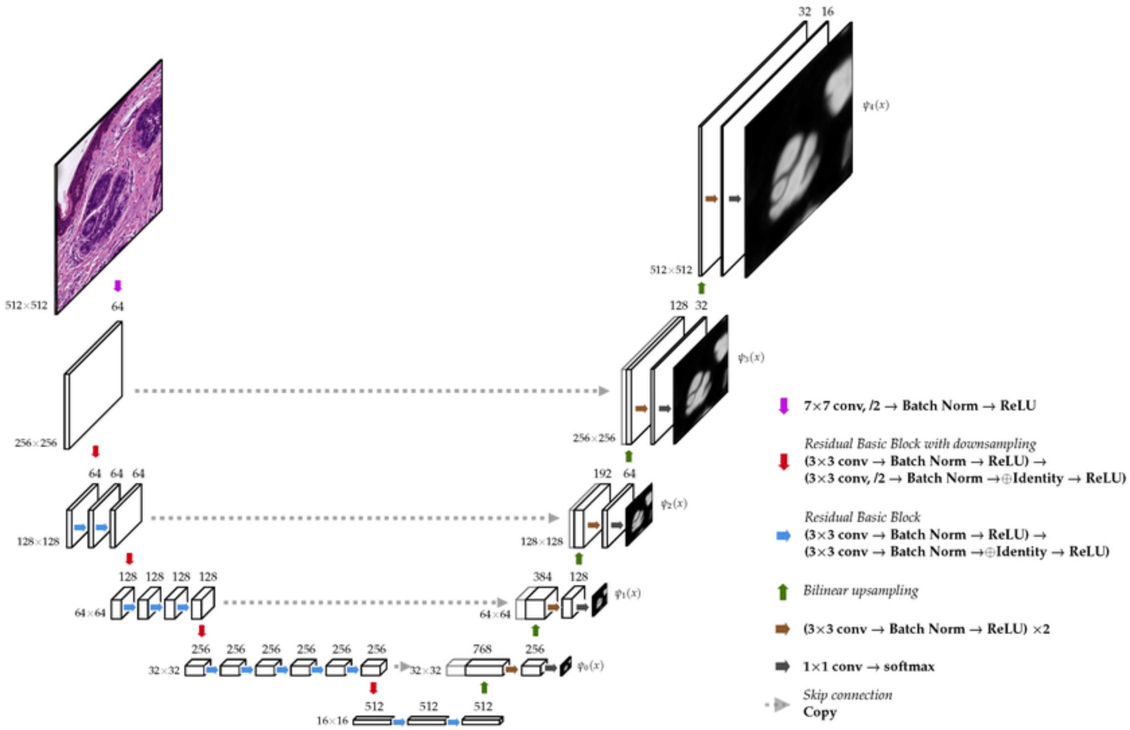

Figure 4: U-Net with Resnet-34 Encoder

Model

The solution involves training a image segmentation model using U-Net architecture [4]. Resnet-34 [1] model is used as encoder. Figure 4 [3] shows this model architecture in detail. The open source package qubvel:segmentation models [2] is used to implement the same. Pretrained imagenet-10k weights are used for initialization of the encoder block.

The input to the model is a 736x736x3 image and it produces 736x736x1 size prediction.The output from the model is compared to the mask provided. Since the mask is binary, a combination of Binary Cross Entropy and Dice loss is used for training, with equal weights to both.

Below is the list of parameters and specifications used for training the model.

Parameter	Value
Model	UNet with Pretrained Resnet-34 Encoder
Optimizer	Adam with learning rate 1e-3 and weigth decay 1e-6
Schedular	ReduceLRonPlateau with Patience 0 and factor 0.1
Epochs	15
Batch Size	6

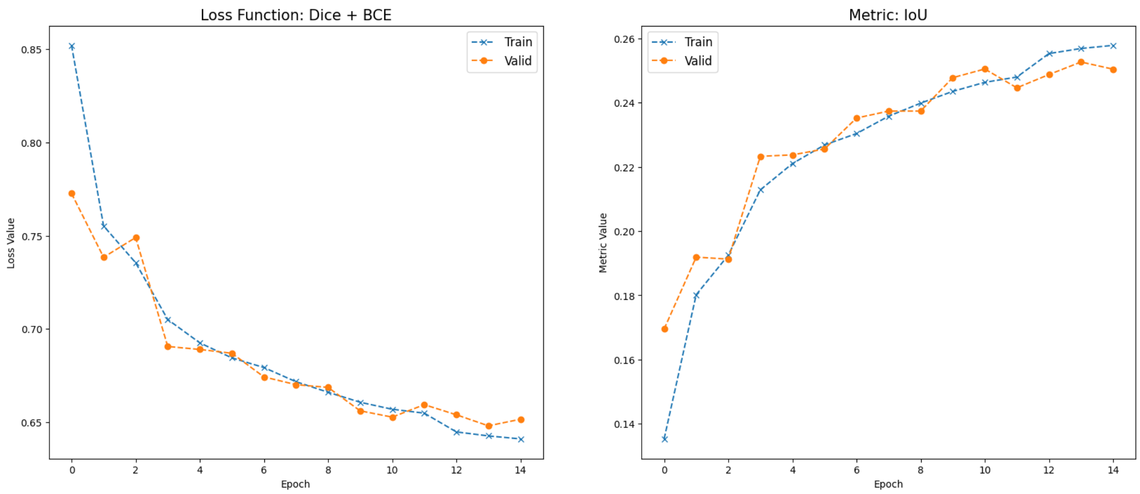

Figure 5: Model Performance

Evaluation

Model is evaluated using IoU (intersection over union) metric. It ranges from 0-1 with higher values denoting better performance. The graphs in figure 5 show how the model performs on this metric as the epochs progress. The final value of IoU on validation dataset is 25.27%. Figure 6 shows how the model performs on a random image from validation dataset after its trained. The first image is the input to the model, the second is the mask provided and the last one being the output generated from the model.

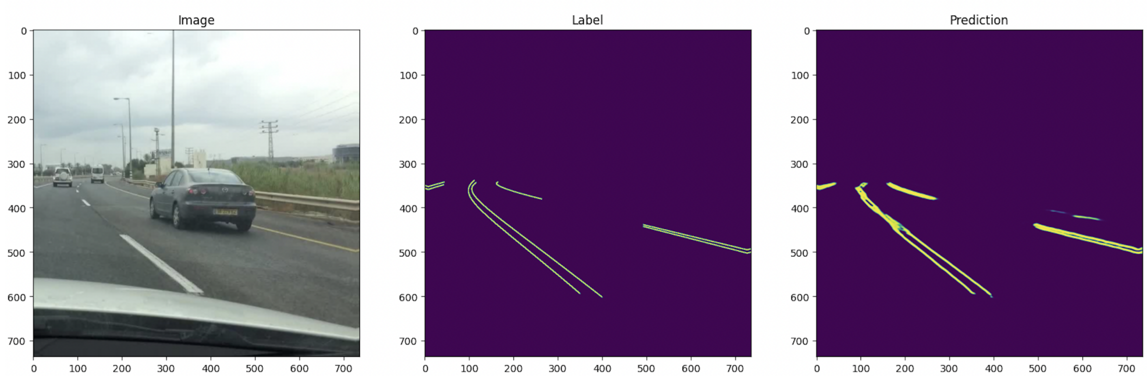

Figure 6: Model Prediction

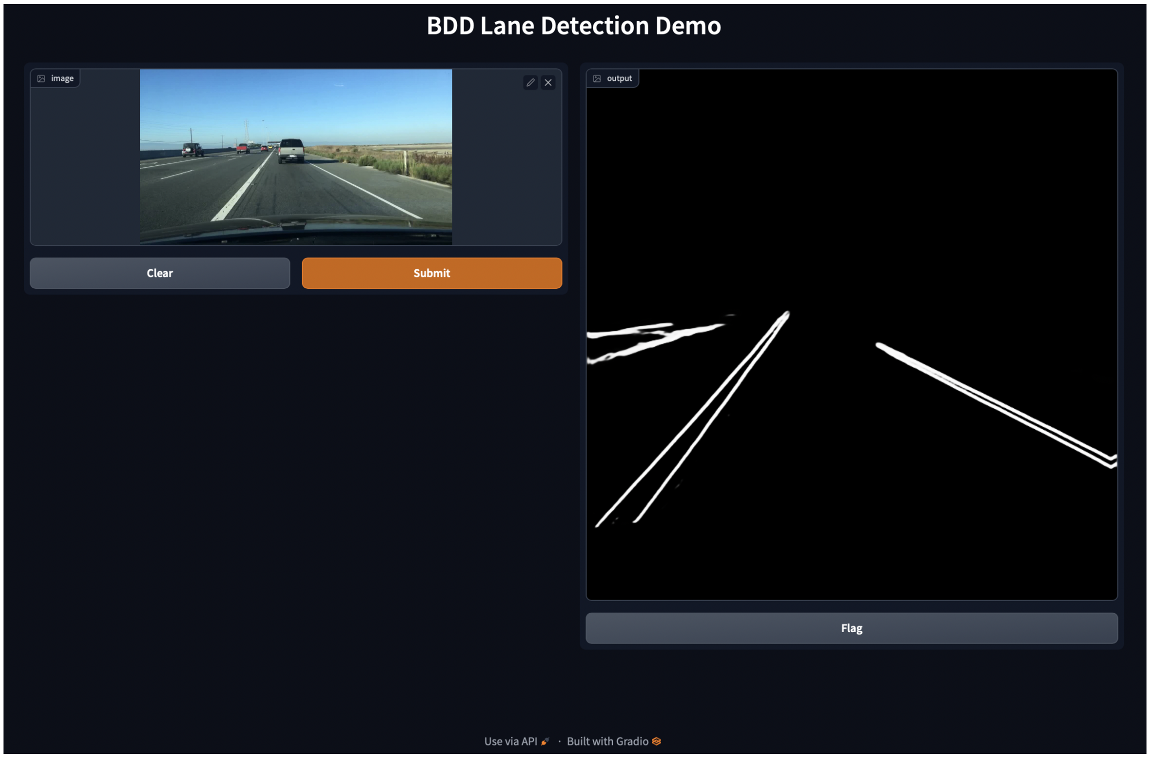

Figure 7: Gradio App Screenshot

Code

- The code for this exercise can be accessed at: Github
- The dataset and trained model files can be accessed at: Kaggle Dataset
- The model can be served using Gradio app as shown in the code repository

References

[1] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. DOI: 10.48550/ARXIV.1512.03385. URL: https://arxiv.org/abs/1512.03385.

[2] Pavel Iakubovskii. *Segmentation Models Pytorch*. https://github.com/qubvel/segmentation_models.pytorch. 2019.

[3] Jean Le'Clerc Arrastia et al. "Deeply Supervised UNet for Semantic Segmentation to Assist Dermatopathological Assessment of Basal Cell Carcinoma". In: *Journal of Imaging* 7 (Apr. 2021), p. 71. DOI: 10.3390/jimaging7040071.

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. DOI: 10.48550/ARXIV.1505.04597. URL: https://arxiv.org/abs/1505.04597.