End to End Freeway Lane Detection using a Semantic Segmentation Approach

Name: Junjie Lou, Xiangbing Ji, Zhengxun Wu
Email: julou@stanford.edu, xji1994@Stanford.edu, wukey92@stanford.edu

INTRODUCTION

Lane detection is to detect lanes on the road and provide the accurate location and shape of each lane. A robust and consistent lane detection engine helps to guide vehicles and could be used in autonomous vehicles and driving assistance system[3].
Here we present a Semantic Segmentation approach with **encoder-decoder** CNN architecture for lane detection.

DATASET

TuSimple Dataset

- 1 second clip of 20 frames, only last frame labeled.
- At most 4 lane markings. (current and left/right lanes)
- Images with corresponding JSON labels in .txt file.
- Each lane's label is represented as a list of points.
- 2.8K 1280×720 images.
- Freeway images only, no local road.

Preprocessing

- Pair each image with a binary label image generated by corresponding labels and resize image to 320×192.
- Randomly shuffle 2.8K images to train(80%), dev(10%), and test(10%) sets.

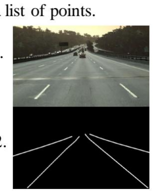


Fig. Image pair

METHOD

Loss function design

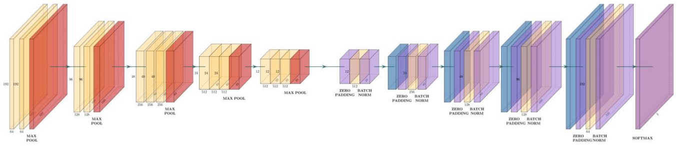
There are way more non-lane pixels than lane pixels (200 : 1). Hence, **weighted cross entropy** is used to compute the loss of pixel classification. β is the weight, as well as a hyperparameter for tuning.

$$\mathrm{WCE}(p, \hat{p}) = -(\beta p \log(\hat{p}) + (1 - p)\log(1 - \hat{p}))$$

Evaluation

We convert the binary prediction image back to the JSON labels. The prediction accuracy is computed as: $accuracy = \frac{\sum_{clip} C_{clip}}{\sum_{clip} S_{clip}}$, where C_{clip} is the number of correct points in the last frame of the clip , S_{clip} is the number of requested points in the last frame of the clip. A point is considered correct if the predicted point and label point is within a certain threshold.

MODEL ARCHITECTURE



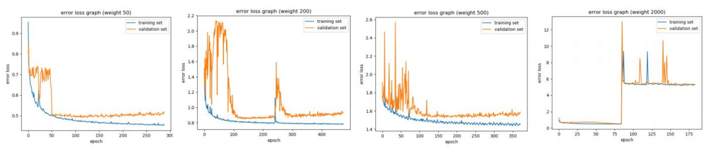
Input

- The model is based on SegNet[1]
- Consists of 11,548,802 parameters in total.
- Consists of an encoder network and a corresponding decoder network.
- The encoder network is identical to the VGG-16.
- The decoder is followed by a pixel-wise binary classifier.

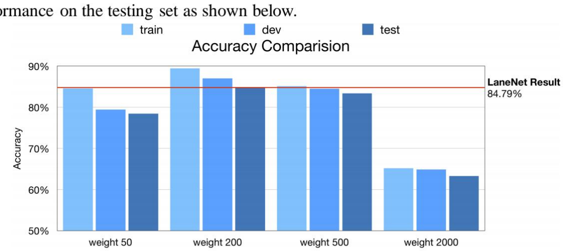
Output

EXPERIMENTS

- We focused on tuning class weights β of values 50/200/500/2000 and plotted their loss graph.

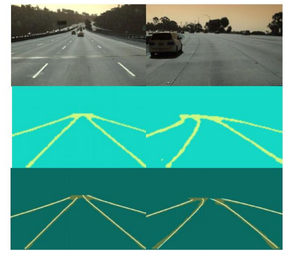


- We chose weights of model using early stopping for each class weight and evaluate the performance on the testing set as shown below.



- We selected the model of class weight 200 to do the error analysis and visualization.

RESULT



Input image

Prediction

Prediction with overlaid ground truth

Check our video demo!

CONCLUSION

We treated lane detection as a semantics segmentation problem and applied an **encoder-decoder** CNN architecture to tackle the problem. We achieved an 84.8% accuracy using TuSimple metrics, which is comparable to some start-of-art models like **LaneNet**. (3rd prize in TuSimple challenge)

FUTURE WORK

- Use loss function with dynamic class weights.
- Collect more data with different scene types.
- Leverage more advanced regularization techniques like Dropout.
- Use more proper interpolation technique when resizing the images.
- Freeze the VGG layers to speed up training.

REFERENCE

[1] Badrinarayanan, et al. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation." (2017)
[2] Neven, Davy, et al. "Towards end-to-end lane detection: an instance segmentation approach." *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018.
[3] Urmson, Chris, et al. "Autonomous driving in urban environments: Boss and the urban challenge." *Journal of Field Robotics* 25.8 (2008): 425-466.