Exploring Semantic Segmentation Problem in Urban Driving Datasets*

1st Komel Merchant *Electrical Engineering*
Johns Hopkins University
Baltimore, Maryland
kmercha2@jh.edu

***Abstract*—This paper explores a binary semantic segmentation problem in the context of urban driving scenes – specifically the problem of car segmentation. In this paper, I explore 2 issue within the space. (1) Is the problem is of dealing with unbalanced datasets. Specifically, I look at the DICE loss and compare performance to the conventional binary cross-entropy loss. 2) Is the question of NN scalability with respect to location. I train a UNet on semantic data from the KITTI [INSERT CITATION HERE] dataset. I then evaluate performance of this dataset on samples from other cities in different countries to see how how well performance scales.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

The area of urban driving is a really important one especially with the rise of autonomous mobility companies[citation here]. Smart cities are also becoming more of a reality in a lot of places. Being able to track vehicle positions over time is crucial for task within robotic perception, traffic estimation and many other technical innovations in urban situations. Vehicle segmentation sits high up in priority within this problem space. In this paper, I explore two main issue within the car semantic segmentation problem.

A. Semantic Segmentation for Urban Driving

u

B. Resolving Class Imbalance

The first is working with unbalanced datasets. In the context of urban scenes, the semantic label space is pretty large. Segmentation algorithms have to interact with roads, poles, signs, cars, sometimes horse-drawn carriage [2]. While cars are a common occurance in images, they take up a smaller proportion of samples in the context of binary segmentation. Furthermore, their frequency is largly governed by other external factors such as time of day, weather conditions, etc. Continuous obstacles such as road ways and such are far less challenging to detect and segment due to their consitency in most urban environments. In the Kitti dataset, the dataset I used for training and evaluation, car's take up approximately 5 percent of the data. If we don't account for this in training our model, our model will learn to produce segmentation masks which produce good general accuracy metrics, but suboptimal performace on the class of interest. An example of this would be the situation where the network might predict a segmentation mask of "not-car". In this case, the accuracy might be high, but the class-specific accuracy will be poor (ie. failure to predit the object of interest: the pixels assoiciated to car). This problem has been a topic of concern in the medical imaging space. In this context, the segmentation masks can potentially be small and localized to different portions of a sample. In the context of tumor segmentation for example, tumors can have varying shapes, sizes and locations within a 3D volume. Furthermore, tumors generally take up a smaller portion of the brain than unaffected brain matter. As such, many researchers in this space have attempted to address these issues, by designing loss functions which weight positive examples more than the negative ones.Milletari [1] propose using a DICE loss, which is similar to Insersection over Union (IoU) loss. For segmentation problems in the multi-label space domain, researchers can weigh certain labels more or less depending on their frequency. Unet **CITATION** does this by using a Wighted Cross Entropy for their training loss.

II. Methods

A. Architecture

For this project, I decided to use the U-Net model. This model was initially used in the context of medical imaging for the use of cell segmentation. The original model features an encoder-decoder pair. The encoder is comprised of a series of blocks which consist of 2 3x3 Convolutions, followed by a standard 2x2 max-pooling layer. There are 4 of these such blocks leading up to the bottleneck. For the decoder portion, there's a series of Up-Convolution blocks that correspond to each block in the encoder. These Up-Convolution blocks consist of 2 3x3 Convolutions, followed by an transposed convolution (also known as a fractional kernel) used to up scale the feature maps. The inputs to these "Up-Conv" blocks are a concatenation of 1) the preceding transposed convolution and 2) a center-crop from the corresponding block in the encoder pair. This cat-crop mirror is what forms the "U" shape in U-Net. Figure 1 shows the original architecture of the model.

I made a few modifications to help with training and testing. The input image for the standard U-Net was 572x572 and output image was 388x388. The authors then used morphological operations to resize the segmentation mask to it's original space. To speed up training time, I resized my input image

to 256x256. Furthermore, I padded my feature maps in the output of all the convolution steps to avoid having to rely on image processing methods to get back to the original size. This simplified validation and loss computations tremendously.

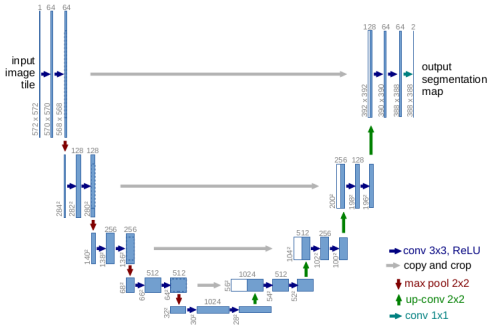

Fig. 1. Figure of the original U-Net.

B. Loss Functions

To address the binary segmentation problem, I experimented with 3 different loss functions: Binary Cross Entropy (BCE), DICE, and a modified version of DICE where I dropped the negative example term.

‘**Binary Cross Entropy**

$$-\frac{1}{N}\sum \tag{1}$$

C. Training

For training, a standard SGD optimizer was used. I used a momentum of 0.9 with an initial learning rate of 0.001. I also added a learning scheduler that’s set to decrease the learning rate by a factor of 10 every time the validation loss remains unchanged for 10 epochs. I set the model to train for 300 epochs. This was more to bound the computatation to a tractable amount, but – as we’ll see in the experiments section – the models’ converged fairly reasonable minima within this time.

D. Datasets

III. Experiments

A. Training and Evaluation on KITTI Semanitic Segmentation Dataset

B. Evaluation on Custom ”DC Commute” Dataset

IV. Discussion

A. Model Performance

Acknowledgment

N/A

References

[1] Brosch, T., Yoo, Y., Tang, L.Y., Li, D.K., Traboulsee, A., Tam, R.: Deep convo- lutional encoder networks for multiple sclerosis lesion segmentation. In: MICCAI 2015. pp. 3–11. Springer (2015)

[2] https://www.reddit.com/r/technology/comments/wt89lv/video$_{a}ppears_{t}o_{s}how_{a}t_{e}slas_{a}utopilot_{s}ystem/$

Fig. 2. Example of a figure caption.