Deep Learning on Images and Signals

U-Nets in a Self-Driving Segmentation Task

Nils Fahrni
BSc Data Science Student

January 17, 2025

Contents

Chapter 1
Task

1.1 Research Question

The research question which I aim to answer with this mini-challenge is:

"How do segmentation models perform between scenes of city streets and non-city streets in the BDD100K dataset?"

This research question is interesting because it addresses the practical challenges of deploying semantic segmentation models in real-world applications, such as autonomous driving, where environmental variability is a key concern. City and non-city environments differ significantly in terms of visual characteristics, object density, and lighting conditions, which can impact the performance of computer vision models.

1.2 Dataset

The BDD100K dataset is the largest driving video dataset, featuring 100,000 videos and supporting 10 tasks for evaluating and advancing multitask learning in autonomous driving. It offers diverse geographic, environmental, and weather conditions, making it a benchmark for studying heterogeneous multitask learning and training robust computer vision models [1].

For this Mini Challenge I use the "10k" subset, which is made up of 10,000 RGB images in a resolution of 1280x720 pixels. Thsee are sampled from the 100,000 videos' frames. This subset is intended for semantic segmentation tasks. These 10,000 images have already been pre-partitioned into a train, validation and test partition. The train partition consists of 8000 images, the validation 1000 and the test 1000. Each image has a corresponding semantic segmentation mask (ground truth) with 19 possible classes.

This smaller subset does unfortunately not have scene attributes but the larger video dataset does. Since the semantic segmenatiton subset is derived from the video dataset, I retrieve the scene attributes through the larger datasets metadata JSON. The issue here is that not all images in the semantic segmentation subset seem to be in the video dataset. I will therefore only use the small overlap of images that has scene attributes and exists both in the video and semantic segmentation dataset. This overlap consists of 3426 images.

1.3 Methodology and Procedure

To answer the research question, I will train a **U-Net** model on the BDD100K dataset. The U-Net architecture is a common choice for segmentation tasks as it contains an Encoder-Decoder structure:

1. **Encoder**: Extracts features from the input image using a series of convolutional and downsampling operations. This is essential in self-driving tasks to distinguish between small objects (like traffic cones) and large areas (like the road).
2. **Decoder**: Gradually upsamples the feature maps and uses convolutions to predict dense segmentation maps. This structure is ideal for segmenting objects in self-driving scenarios, such as lanes, vehicles, pedestrians, and road signs.

As a second model I will modify the U-Net model to include an attention mechanism, which I will implement myself according to [2]. I expect the addition of attention to be another improving factor because:

1. **Driving Reality**: Different objects and their spatial relationships often define the context. For instance:

(a) A cyclist is more likely to be found near a bike lane or the edge of a road.

(b) A pedestrian might be near a crosswalk but not in the middle of a highway. Cars and trucks are expected on roads but not sidewalks.

2. **Attention Benefit**: The attention mechanism allows the model to focus not just on isolated objects but also on the relationships between them. It helps the network infer that the presence of a bike lane increases the likelihood of a cyclist or that highway lanes imply the absence of pedestrians and cyclists.

I will then evaluate the models on the test set and compare the performance between city street and non-city street scenes. To measure and evaluate the performance between different model complexities numerically, I will use the mean Intersection over Union (mIoU) as the evaluation metric.

Both models will be explored within the schema of "A Recipe for Training Neural Networks" according to Andrej Karpathy [3].

Chapter 2
Notable Results

In this chapter I will present specifics of my two best models, once the Vanilla U-Net that has reached the best IoU scores during Overfitting and once the Attention U-Net that was explored in the last phase of the project (Section 3.3.4 "Squeeze The Juice").

2.1 Vanilla U-Net

The Vanilla U-Net with `128` base filters, projecting into a latent space of `2048` dimensions has reached the best IoU scores during the Overfitting phase.

2.1.1 Training

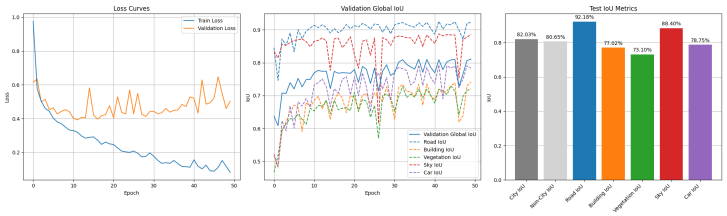

Figure 2.1: Training and Validation of the Vanilla U-Net with 128 Base Filters.

The experiment yielded the desired result of **overfitting**. Though the implemented trainer always saves the model at its **lowest validation loss**, thats where best generalization was reached.

2.1.2 Sampled Results

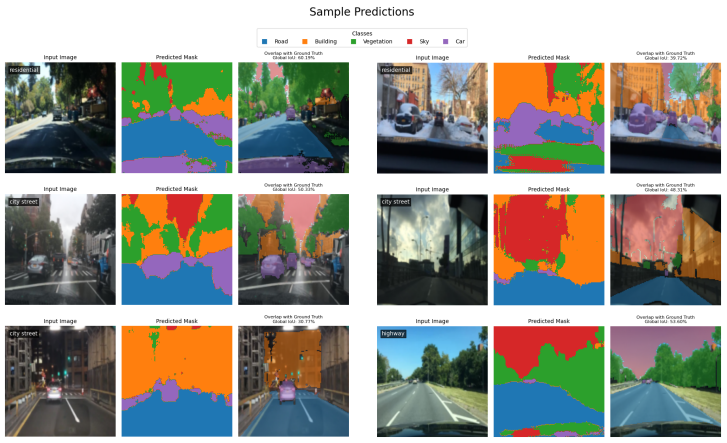

Figure 2.2: Difficult Samples predicted by the Vanilla U-Net

The rather simple Vanilla U-Net also performs surprisingly well when taking a visual inspection at the results of some difficult samples. Both the `road` and `car` classes are predicted well. There are some samples where the model seems to struggle, for example in snowy scenes where the road is not entirely visible.

2.2 Attention U-Net

The Attention U-Net essentially is a Vanilla U-Net but with Attention Blocks built in. In the beginning experiments with the Attention U-Net I observed that it did started to overfit quite quickly due to the added parameters. This is when I decided to explore adding dropout and found a dropout rate of 20% to be beneficial.

2.2.1 Training

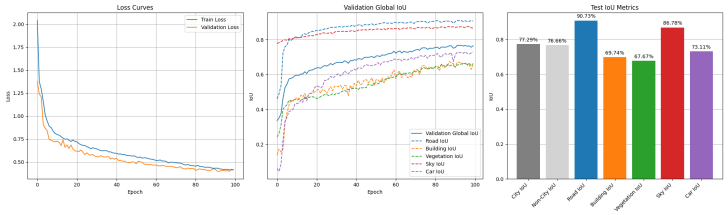

Figure 2.3: Training and Validation of the Attention U-Net with Dropout.

Contrary to the Vanilla U-Net's loss curves we can now observe in the Attention U-Net that the validation loss is decreasing along with the training loss. This is a good sign that the model is not overfitting and starting to generalize well since the validation loss starts to converge.

2.2.2 Sampled Results

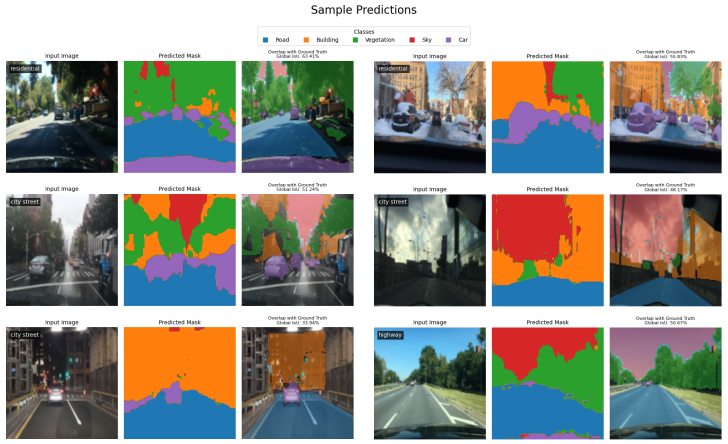

Figure 2.4: Difficult Samples predicted by the Attention U-Net

Even though the numerical metrics of the Vanilla U-Net are slightly better, the Attention U-Net seems to perfomr better in the samples where the best Vanilla U-Net still had some difficulties. For example, in the first sample, the Attention U-Net gets an even better understanding of smaller segments and labels that lie in the shadow. This is also observable in the fourth sample where the road is covered in snow.

2.2.3 Saliency Maps

As a last step in the Attention Mechanism exploration I also wanted to look at the saliency maps of the attention mechanism.

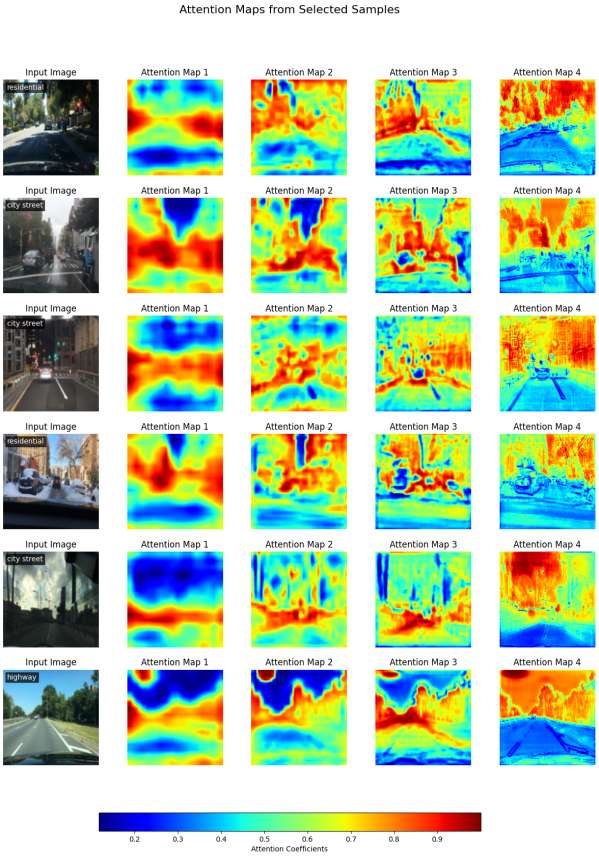

Figure 2.5: Saliency Maps of the Attention Mechanisms ψ Coefficient.

The first attention layer seems to focus on the general location of the objects, it especially attends to the sides of the streets. This could be due to the fact that often times the sides are the biggest discriminator for what will be in a scene. For example, in the second sample, the image has a lot of different objects. The attention layer at that point attends to all objects on the sides where compared with the last sample, the attention layer does not have too many objects that are "hard to classify" as the scene is mostly just `road` and `sky`. Though at this last sample, there is an interesting activation happening at the top left of the image - I can't really make out why that is but perhaps it gets some information from the image having clear skies at this point.

The second layer goes more into detail and focuses on clearer shapes, this is a pattern that can be observed in the following attention maps as well. At attention map 4 we can see the mechanism has clear shapes to attend on. The ψ coefficient seems to be especially high for parts of the image with a lot of structure as that is where it can gather a lot of information for labeling.

Chapter 3
Discussion

3.1 Answering The Research Question

To answer if the models answer the research question in a satisfactory manner I have calculated the Intersection over Union (IoU) metric for both city scene images and non-city scene images from the held out test datset. This allows me to compare the main metric between all explored models. Here I show the winner models from all explorations in the training process:

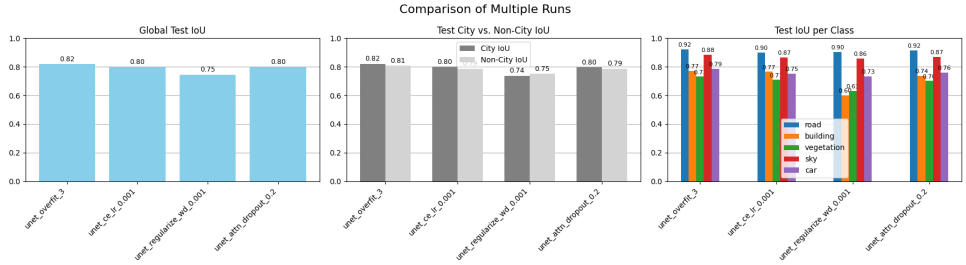

Figure 3.1: Comparison of the winner models.

The center subplot shows the mentioned comparison of non-city scene and city scene IoUs. When it comes to numerical analysis exclusively, the Vanilla U-Net (here, named `unet_overfit_3`) performed the best. This model is expected to overfit, however only the lowest validation loss weights get saved, thus, we do not necessarily see "bad" results as the model was saved at a state where it generalizes well.

Upon further visual inspection, the Attention U-Net with Dropout regularization also performed quite well. It has has slightly lower IoU metrics than the Vanilla U-Net with `128` base filters (`unet_overfit_3`) but seems to visually get more details correct in more difficult environments.

To answer the research question, the models are undoubtedly capable of performing well in both types of scenes. There is no significant difference between both recorded metrics. Thus, the models are not clearly biased towards one scene or have a clear deficit towards one direction.

3.2 Chances and Risks

Here I will discuss the challenges and risks of the proposed models and the approach behind them:

Chances

+ With this mini challenge I present a clear framework of components that would allow for a streamlined exploration of further models and hyperparameters.
+ Looking at the results, the models did succeed in the most critical classes like `road` and `car` which are essential for systems that require lane holding and collision avoidance.
+ The models at hand are derived from a well-known architecture, the U-Net which is simple but yet reached quite good results.

Risks

- The models were trained on a limited dataset, with less than 3500 images and only the 5 largest classes.
- Sampling from the models can take a long time with the larger models proposed which is not ideal for real-time applications like self-driving.
- The handling of classes in the images that were removed for training were not labeled into a clear "background class" so the model learned to assign one of the 5 most common classes to those objects.

3.3 Comparing The Models

To conclude the steps that included training of the Machine Learning Recipe proposed by Karpathy [3], I compared all models of their corresponding step. Here are these interim results:

3.3.1 Overfitting Experiments

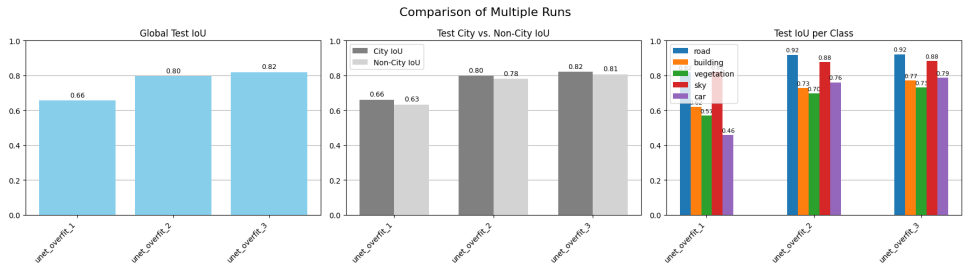

Figure 3.2: Comparison of the models explored in the Overfitting Phase.

In the overfit experiments I have gradually increased complexity by adding more base filters to the Vanilla U-Net. The results showed that the model benefits from a higher complexity - The highest increase happened between `32` and `64` base filters. The model with `128` base filters performed the best in the overfitting experiments.

3.3.2 Regularization Experiments

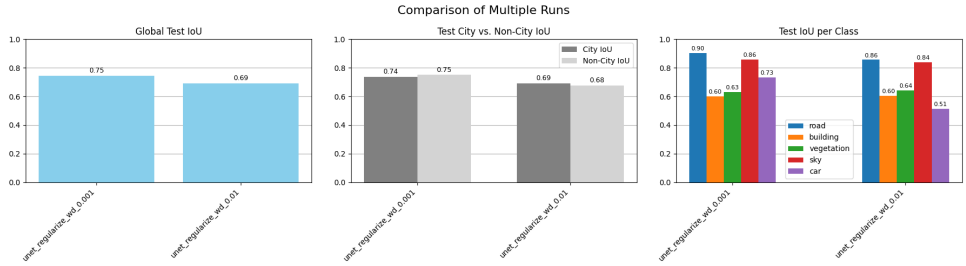

Figure 3.3: Comparison of the models explored in the Regularization Phase.

During regularization I only explored different `weight_decay` parameters. I believe there is still some potential to explore other regularization techniques with the Vanilla U-Net. I have later on explored Dropout regularization with the Attention U-Net which looked more promising in regards to the loss curves.

3.3.3 Hyperparameter Tuning

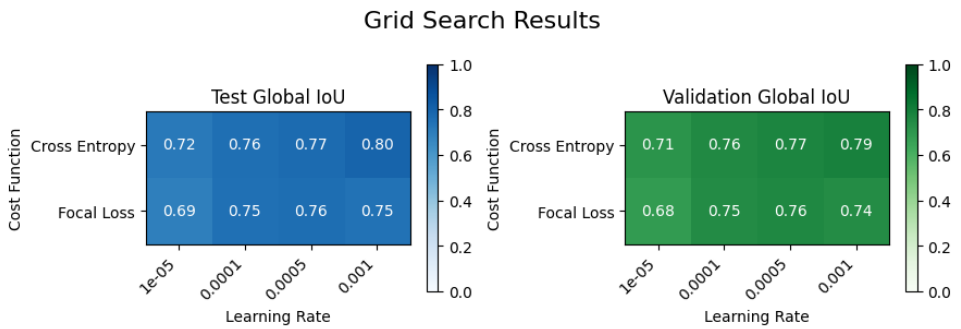

Figure 3.4: Comparison of the models that went through Grid Search.

I also wanted to implement the Focal Loss error function so that presented itself as a good hyperparameter to explore. Ultimately, what resulted was that the Focal Loss may be too nuanced for this dataset as there is no acute imbalance between the classes so a more classical loss function like the Weighted Cross Entropy turned out to work better.

3.3.4 Squeeze The Juice: Attention Mechanism

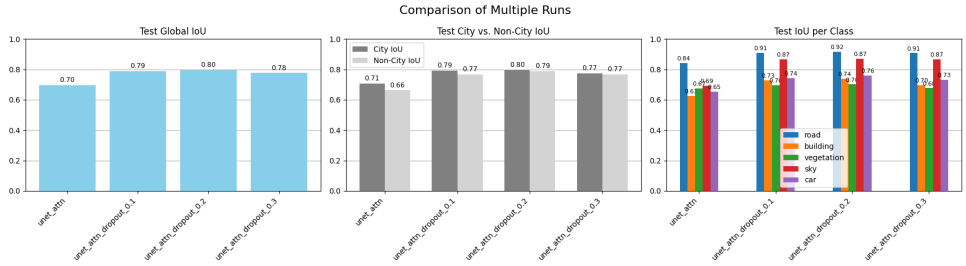

Figure 3.5: Comparison of the Attention U-Net Models.

At last I also wanted to look at what impact implementing the Attention mechanism could have. There have been some publications like "Crack Semantic Segmentation using the U-Net with Full Attention Strategy" by Lin et al. [4] showing that the attention mechanism can have meaningful impact on the performance of semantic segmentation models. My results did reach a similar IoU ceiling for all recorded scores but upon further visual inspection showed the real strenghts of attention.

Chapter 4
Reflection

Reflecting on the work, I found that the dataset presented significant challenges, making the modeling process particularly demanding. Reducing the number of classes provided a meaningful performance boost, yet the presence of numerous "falsely" labeled objects remained a major hurdle, especially for simpler models trying to capture the finer nuances. Despite these challenges, the Attention U-Net exceeded my expectations. Considering the limited data available, I was impressed by how well it performed—especially since attention mechanisms are typically data-hungry. The addition of dropout regularization played a crucial role in further enhancing the Attention U-Net's ability to generalize.

Although the bare metrics suggested that the Attention models performed slightly worse, a closer examination of selected samples convinced me otherwise. The Attention mechanism truly helped the model focus on critical parts of the image, excelling in complex scenes where clarity and confidence in object labeling were vital.

Looking ahead, I see room for further improvements:

- Augmenting the dataset to increase the number of training samples
- Refining the dataset's ground truths by addressing falsely labeled objects, perhaps by leveraging third-party annotations

These steps could help overcome some of the key challenges encountered and further enhance the performance and reliability of the models.

Bibliography

[1] F. Yu *et al.* "BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning." arXiv: 1805.04687, Accessed: Nov. 25, 2024. [Online]. Available: http://arxiv.org/abs/1805.04687, pre-published.

[2] O. Oktay *et al.* "Attention U-Net: Learning Where to Look for the Pancreas." arXiv: 1804.03999, Accessed: Dec. 3, 2024. [Online]. Available: http://arxiv.org/abs/1804.03999, pre-published.

[3] A. Karpathy. "A Recipe for Training Neural Networks," Accessed: Dec. 3, 2024. [Online]. Available: https://karpathy.github.io/2019/04/25/recipe/.

[4] F. Lin, J. Yang, J. Shu, and R. J. Scherer. "Crack Semantic Segmentation using the U-Net with Full Attention Strategy." arXiv: 2104.14586 [cs], Accessed: Jan. 17, 2025. [Online]. Available: http://arxiv.org/abs/2104.14586, pre-published.