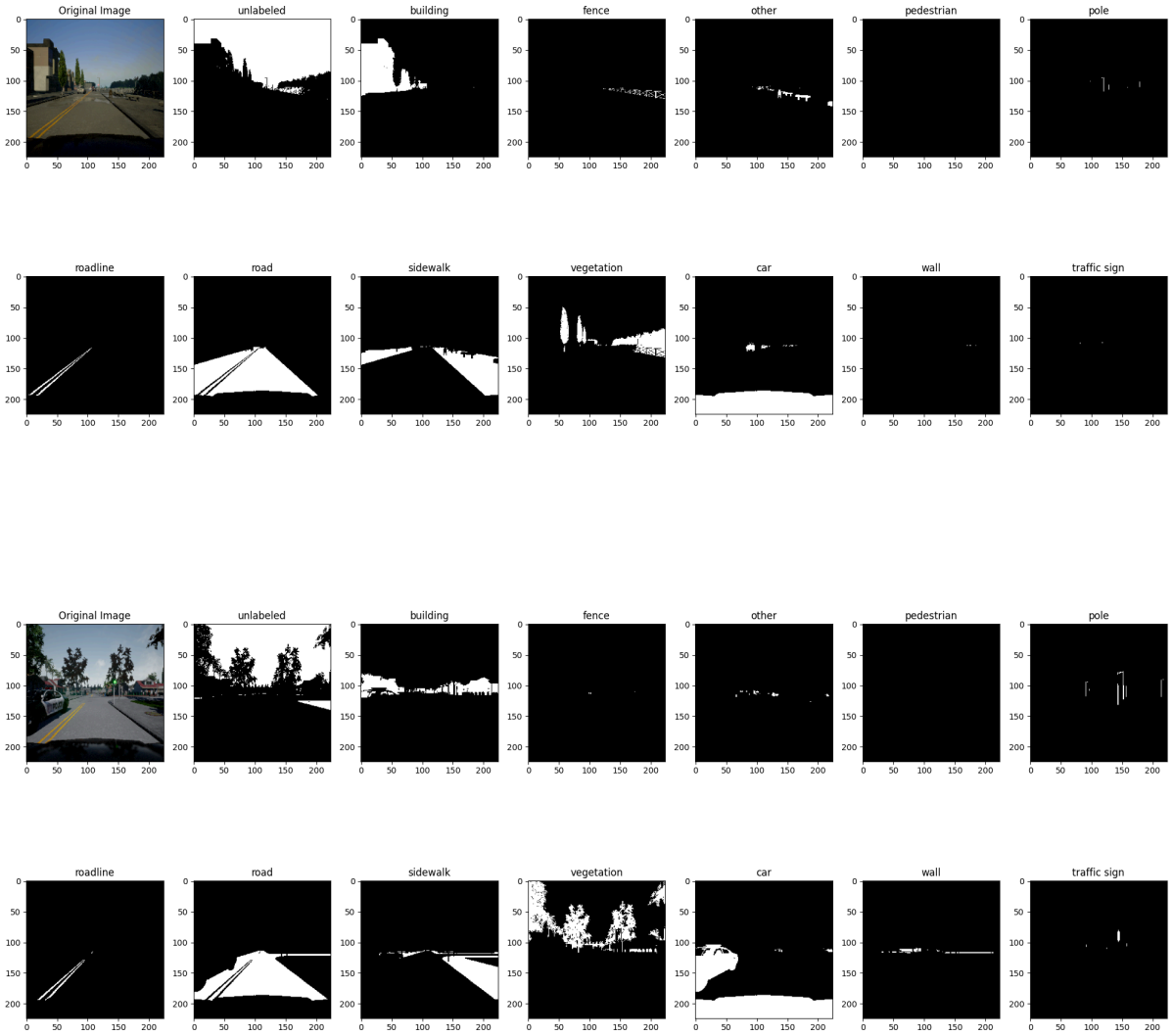
Assignment 4 Report

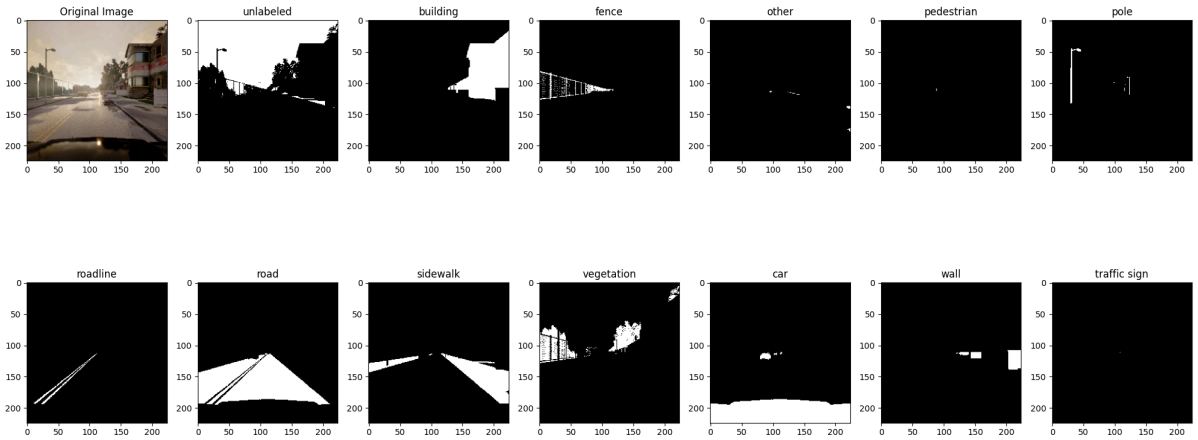
2021101058

Q1

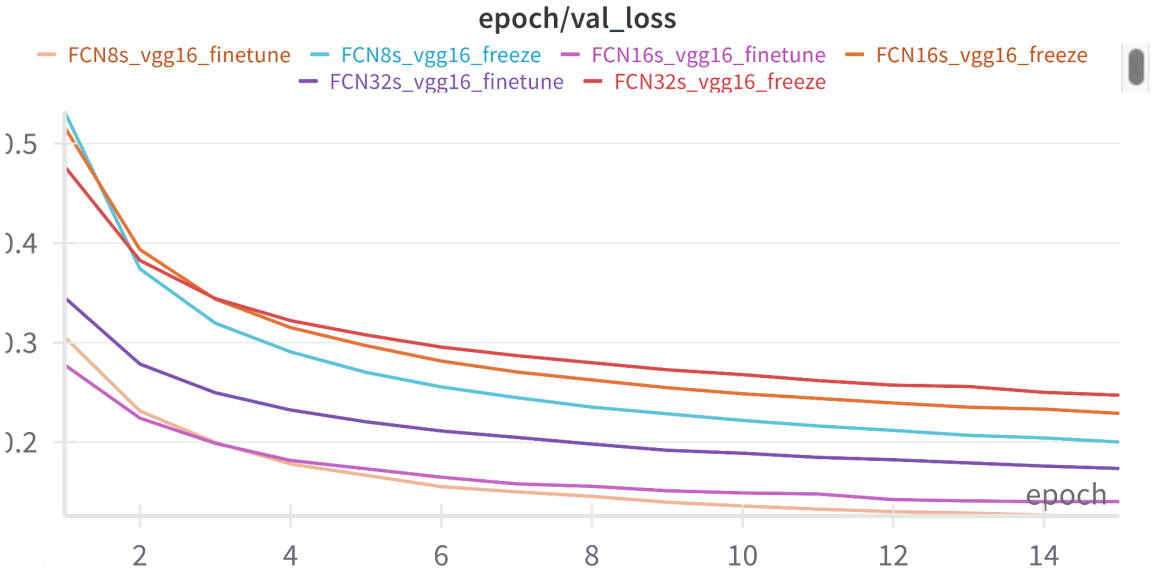
Class Specific Visualizations

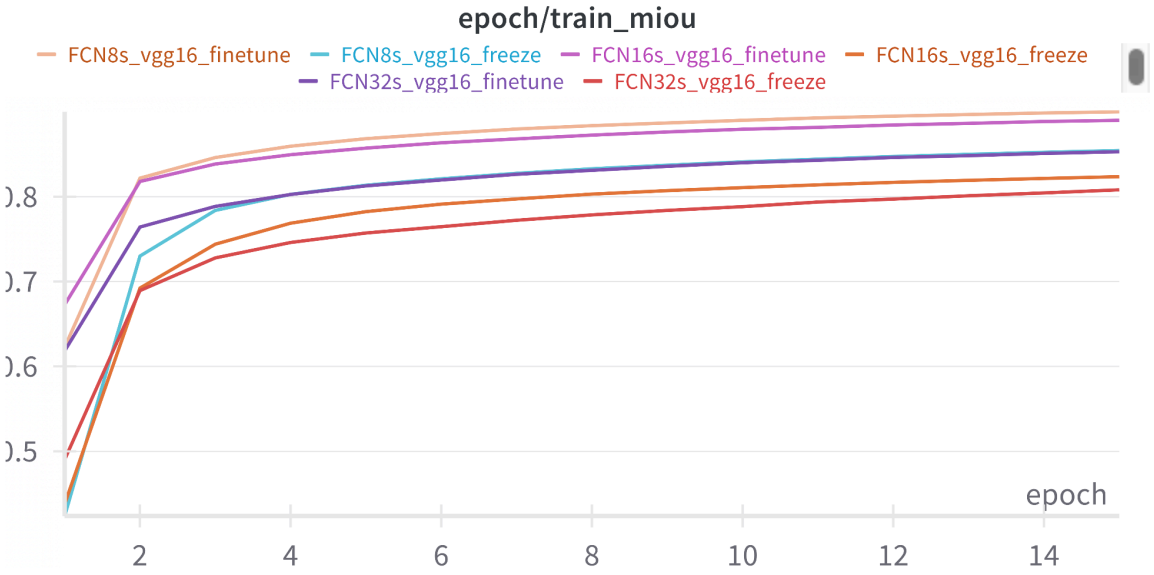
See q1/data_visualization.ipynb for implementation

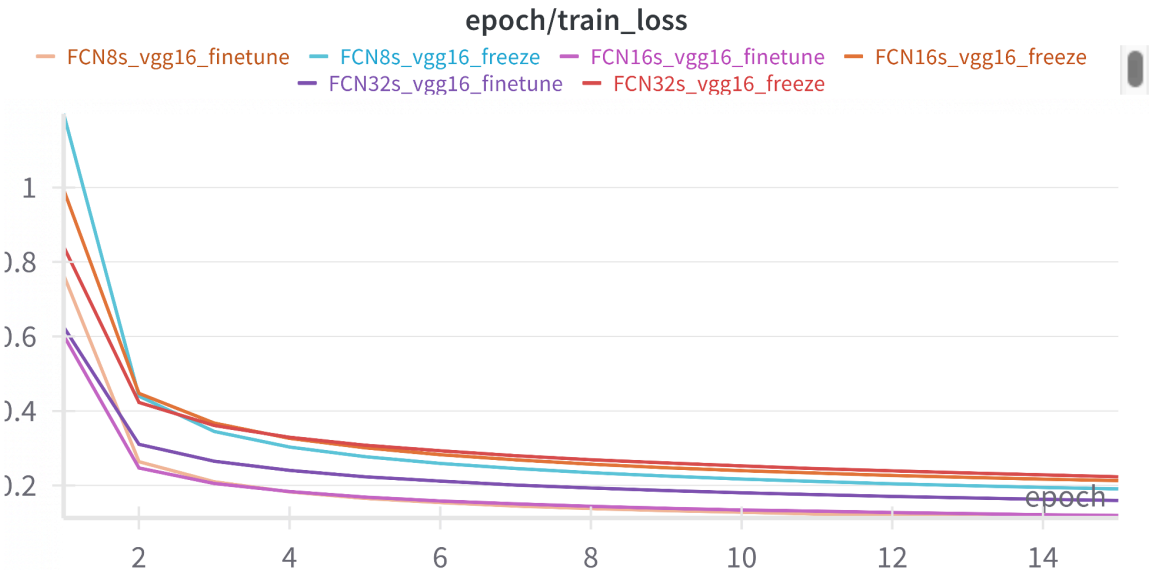


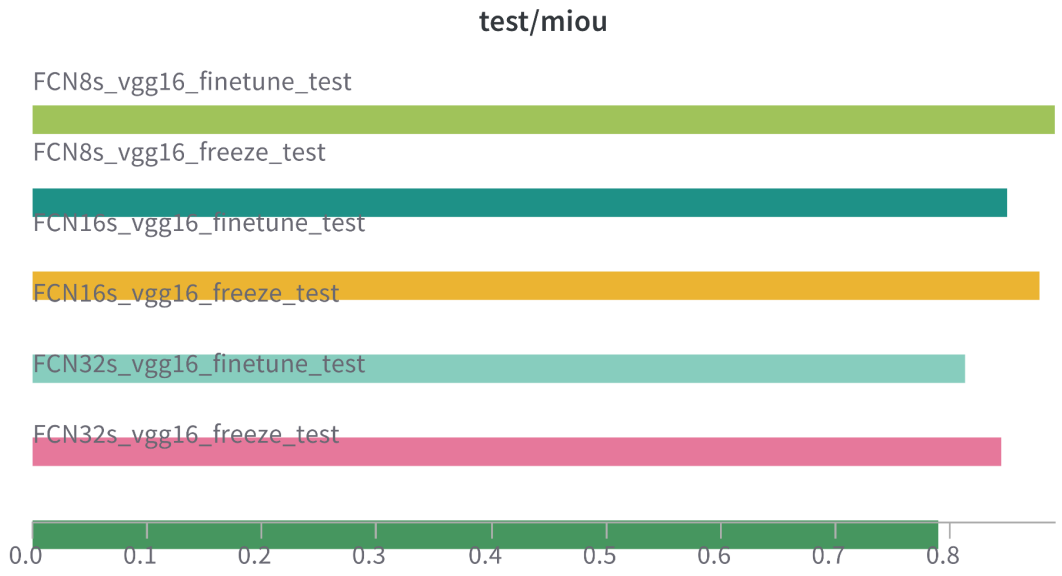


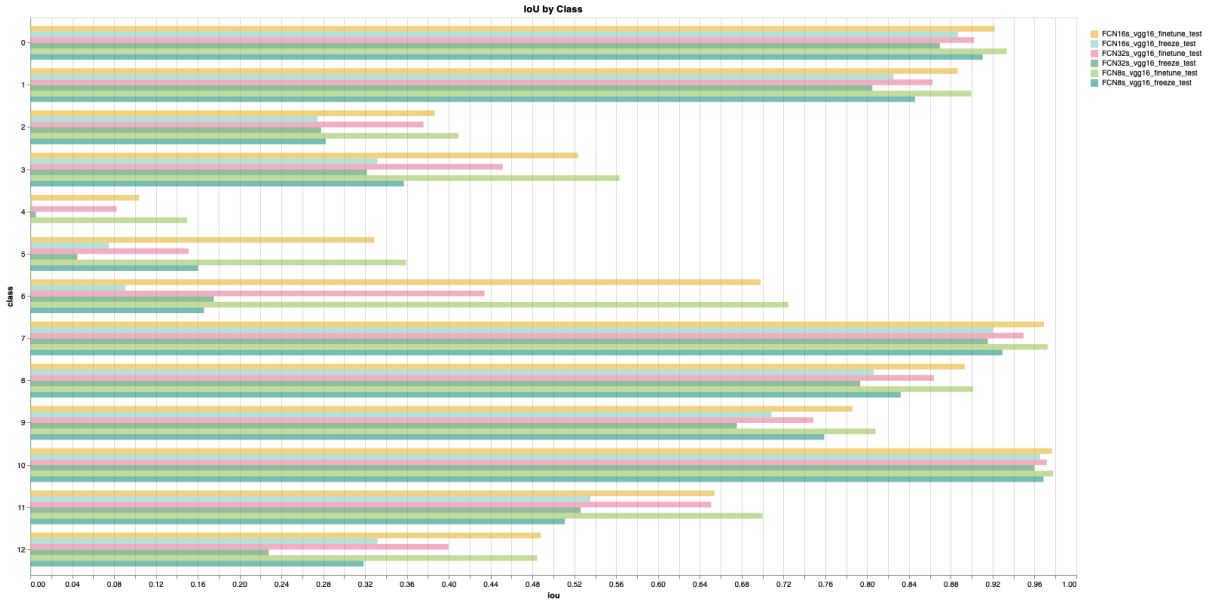
Training and Results Graphs

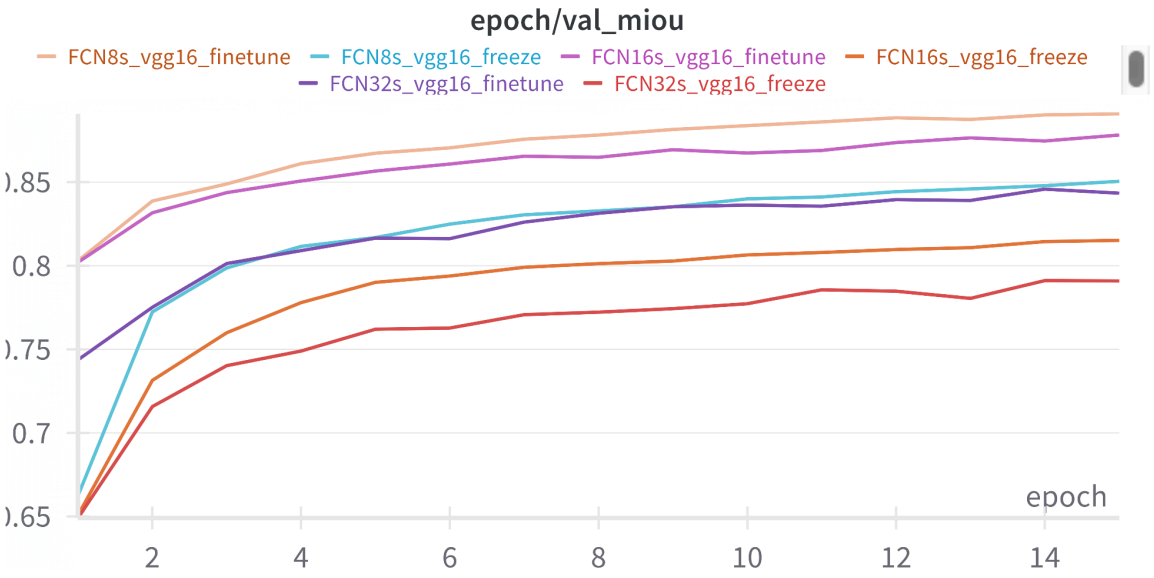

epoch/val_loss
FCN8s_vgg16_finetune
FCN8s_vgg16_freeze
FCN16s_vgg16_finetune
FCN16s_vgg16_freeze
FCN32s_vgg16_finetune
FCN32s_vgg16_freeze
0.5
0.4
0.3
0.2
epoch
2
4
6
8
10
12
14


epoch/train_miou
FCN8s_vgg16_finetune
FCN8s_vgg16_freeze
FCN16s_vgg16_finetune
FCN16s_vgg16_freeze
FCN32s_vgg16_finetune
FCN32s_vgg16_freeze
0.8
0.7
0.6
0.5
epoch
2
4
6
8
10
12
14


epoch/train_loss
FCN8s_vgg16_finetune
FCN8s_vgg16_freeze
FCN16s_vgg16_finetune
FCN16s_vgg16_freeze
FCN32s_vgg16_finetune
FCN32s_vgg16_freeze
1
0.8
0.6
0.4
0.2
epoch
2
4
6
8
10
12
14

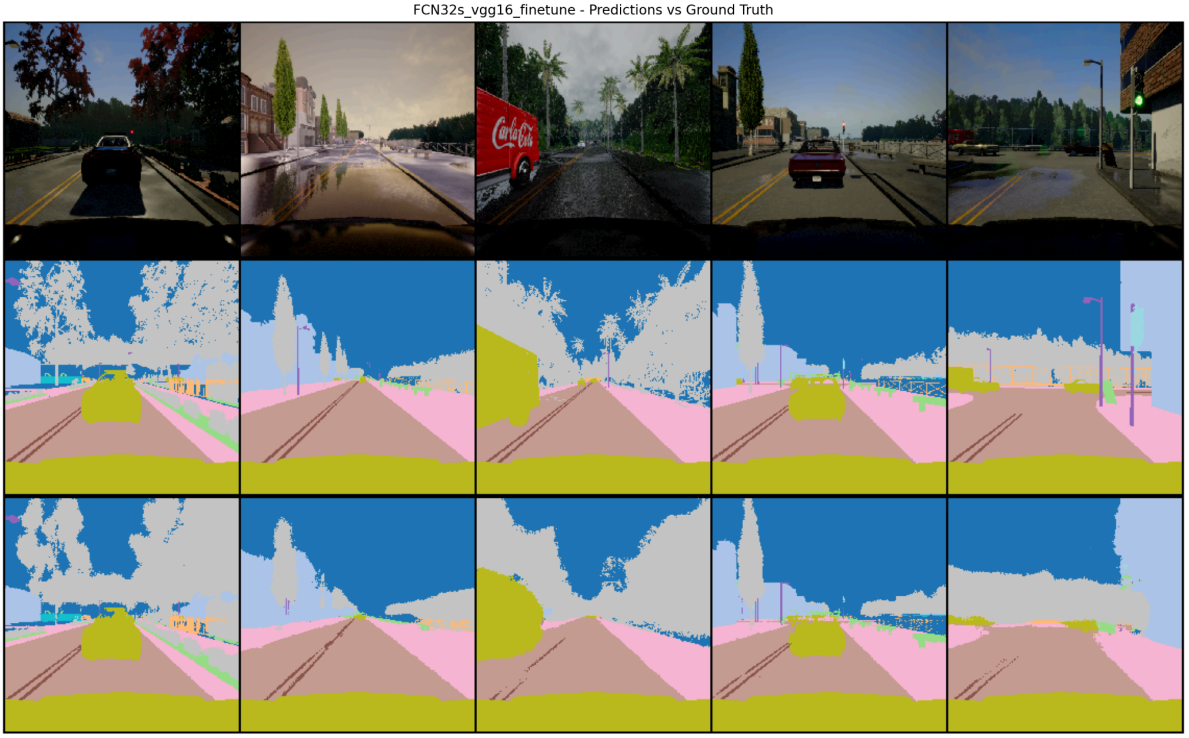

test/miou
FCN8s_vgg16_finetune_test
FCN8s_vgg16_freeze_test
FCN16s_vgg16_finetune_test
FCN16s_vgg16_freeze_test
FCN32s_vgg16_finetune_test
FCN32s_vgg16_freeze_test
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8



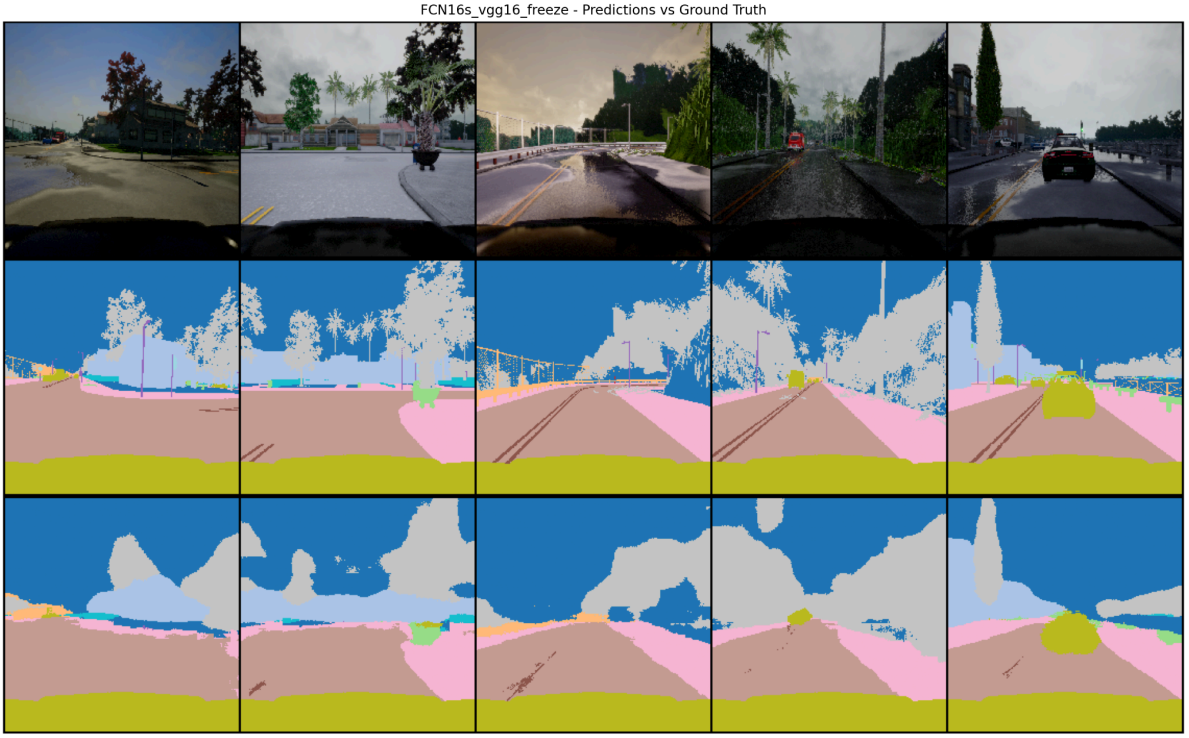


Visualizations:

FCN32s_vgg16_finetune - Predictions vs Ground Truth

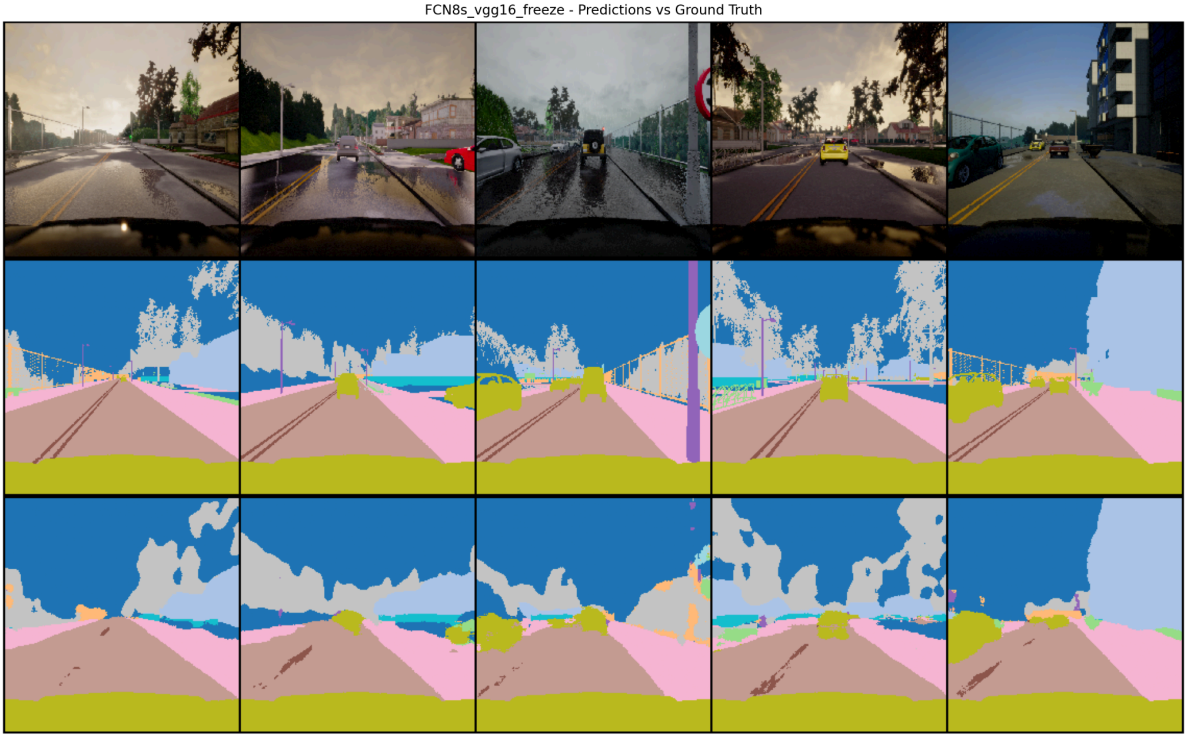


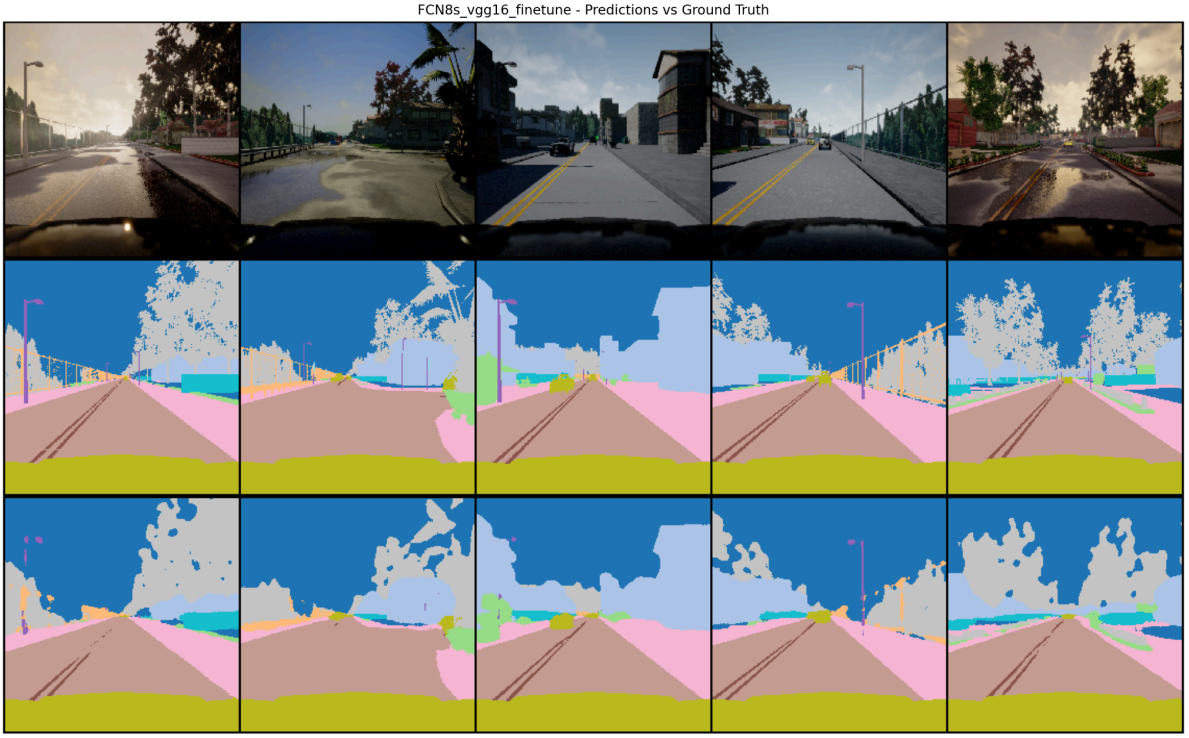
FCN16s_vgg16_freeze - Predictions vs Ground Truth



FCN16s_vgg16_finetune - Predictions vs Ground Truth

FCN8s_vgg16_freeze - Predictions vs Ground Truth





Differences Between FCN Variants

Architecture:

The FCN family includes three main versions—FCN-32s, FCN-16s, and FCN-8s—each building on the last by adding more detail and better handling of spatial information through skip connections:

•FCN-32s is the most basic. It just takes the output from the deepest layer (called pool5) and upsamples it by 32× to match the original input size. Since it doesn't bring in any information from earlier layers, its segmentation results are quite rough and often miss the fine details, especially around object boundaries.

•FCN-16s improves on that by introducing a skip connection from the pool4 layer, which holds more spatial detail. It first upsamples the pool5 output by 2×, combines it with pool4, and then upsamples again by 16×. This extra layer of detail helps it produce cleaner, more accurate segmentation maps compared to FCN-32s.

•FCN-8s goes even further by also pulling in features from the pool3 layer. After merging pool5 and pool4 like in FCN-16s, it upsamples that intermediate result by 2×, adds in pool3, and finally upsamples 8× to get back to the original size. Thanks to these higher-resolution features, FCN-8s achieves the most precise segmentation, especially when it comes to outlining object boundaries.

Segmentation Performance:

In practice, FCN-8s consistently performs the best out of the three. It benefits from using more detailed features from earlier in the network, which helps preserve spatial accuracy.

Looking at how each one behaves:

•FCN-32s often creates blurry or skewed segmentations. It's okay with big, obvious objects but tends to mess up the edges and completely misses smaller objects.

•FCN-16s is a step up—it handles medium-sized objects better and gives smoother boundaries, though it can still struggle with the smallest details.

•FCN-8s shines here. It gives the most accurate, detailed results, doing a good job of detecting both small and medium-sized objects, with crisp boundary outlines.

Quantitatively, we can see this reflected in their mean Intersection-over-Union (mIoU) scores, which steadily improve from FCN-32s to FCN-16s to FCN-8s when trained under the same settings.

———

Frozen vs. Fine-Tuned Backbones:

All these FCN models typically use a backbone network like VGG16 that's been pre-trained on ImageNet. When training, you have two options: keep the backbone frozen (i.e., don't update its weights) or fine-tune it along with the rest of the network.

•Freezing the Backbone means training is faster and uses less memory, but the downside is that the features were learned for classification, not segmentation. As a result, the performance may be limited, especially in pixel-level prediction tasks.

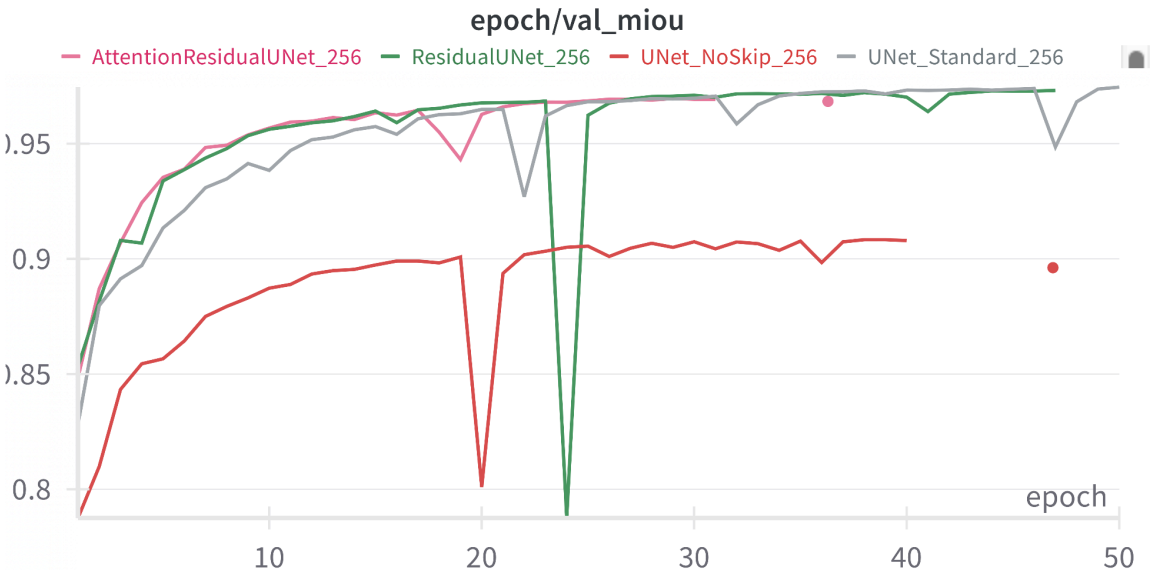
•Fine-tuning the Backbone lets the network adapt those features specifically for segmentation. While it takes more time and compute, it usually leads to noticeably better results—particularly when your data looks very different from ImageNet (like simulation environments, for instance).

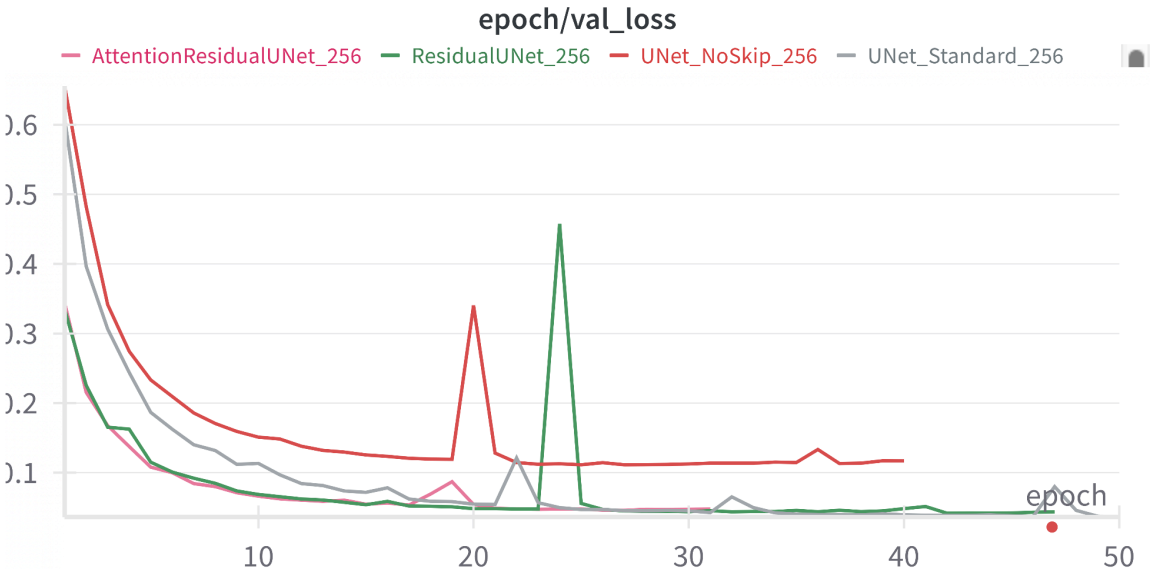
In practice, the benefits of fine-tuning aren't huge for FCN-32s and FCN-16s, especially early in training. But with FCN-8s, the improvement is much clearer—fine-tuning helps it make the most of those high-resolution features, which really matters when trying to get precise, detailed segmentations.

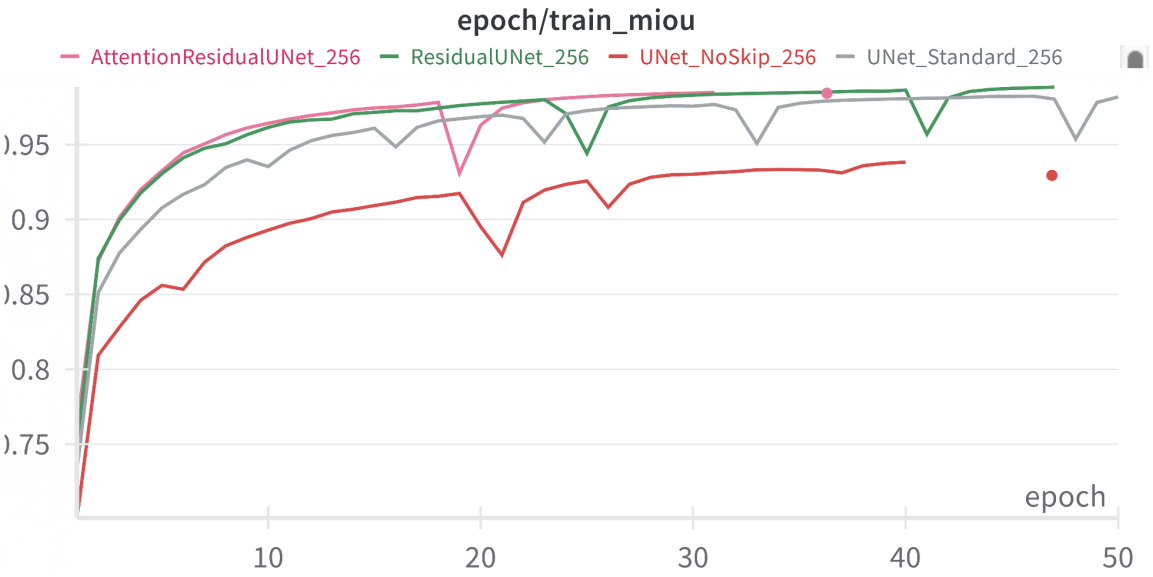
Q2

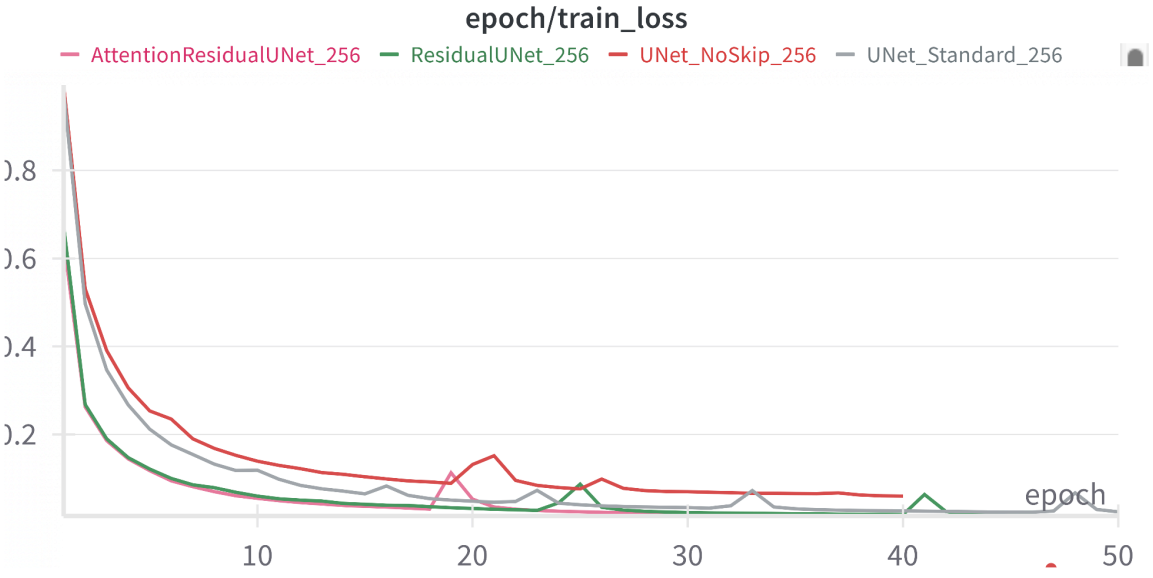
Attention gate 1: conv becomes 2x2 with stride 2 to get both to the same dimension before element-wise addition
Attention gate 2: Passing the gating signal after upconv

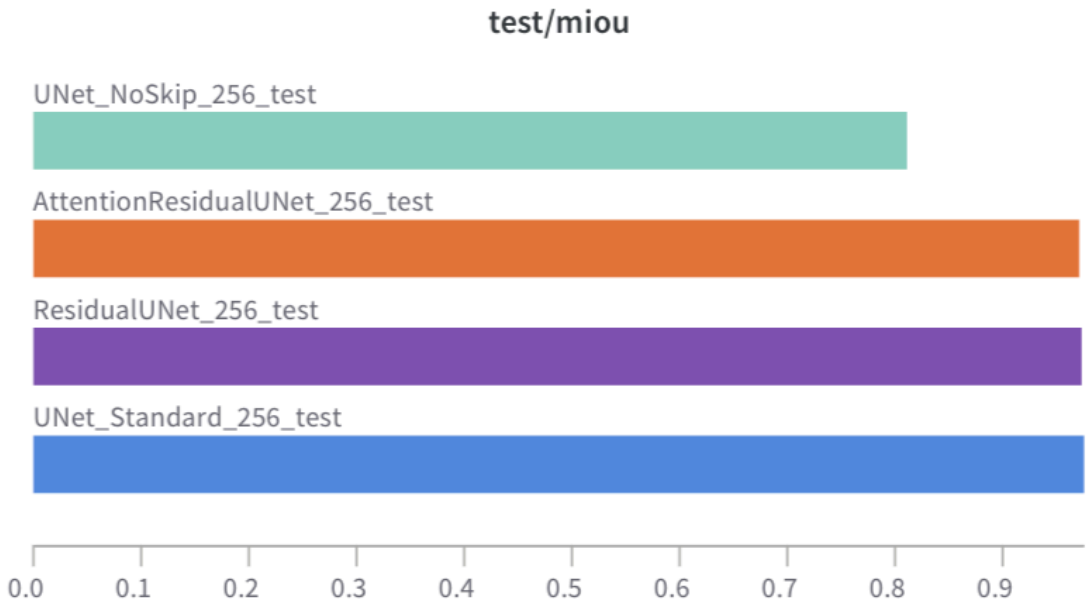
Training and Results Graphs

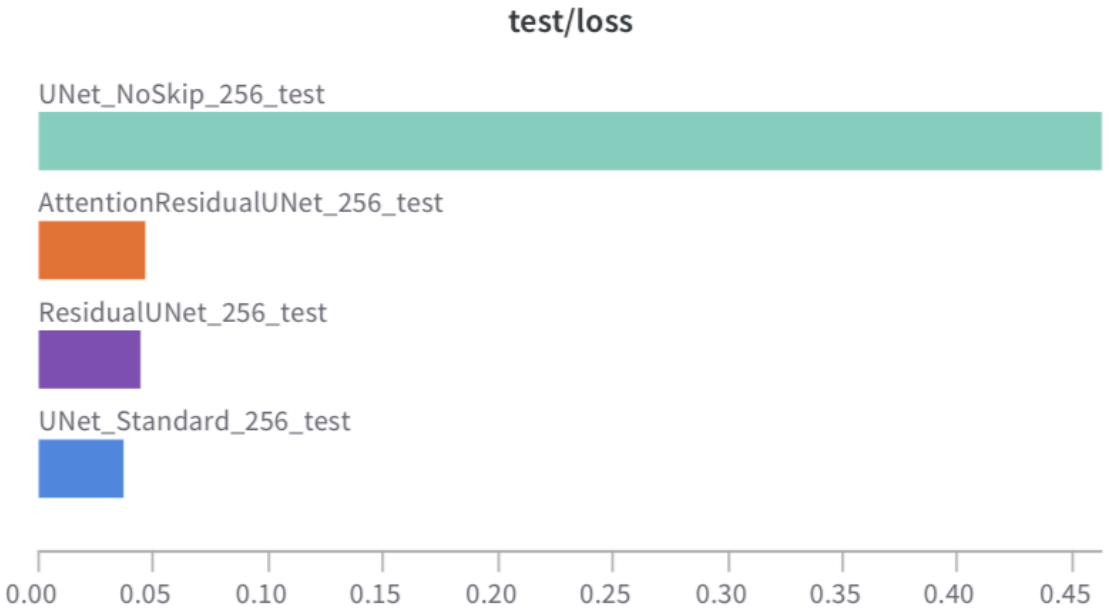


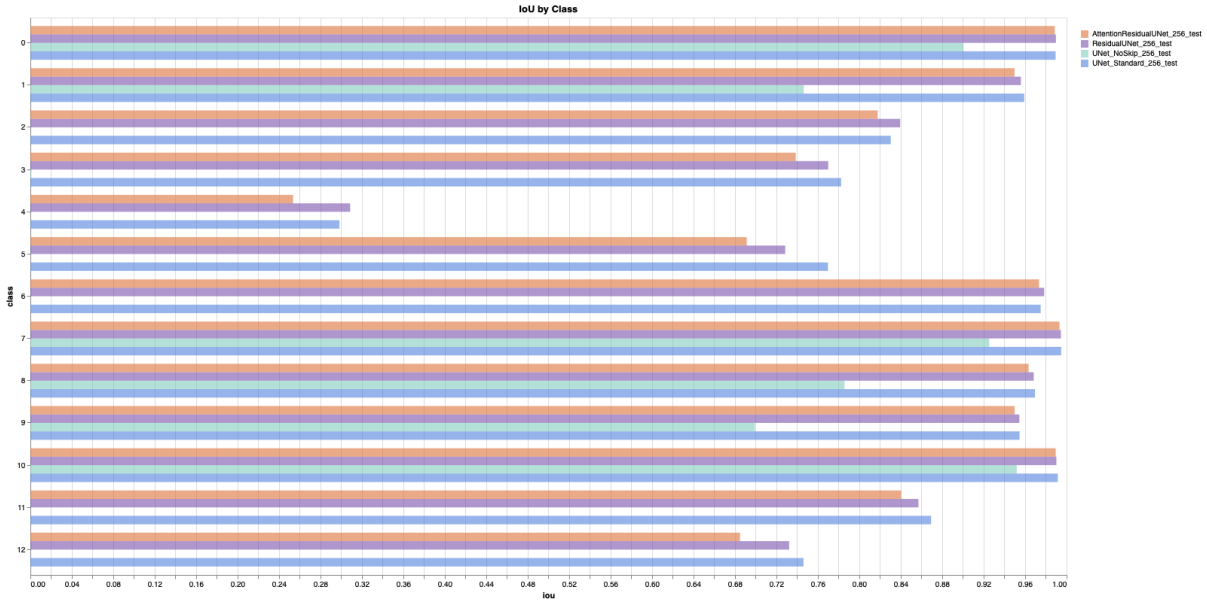



epoch/train_miou
AttentionResidualUNet_256
ResidualUNet_256
UNet_NoSkip_256
UNet_Standard_256
0.95
0.9
0.85
0.8
0.75
epoch
10
20
30
40
50


epoch/train_loss
AttentionResidualUNet_256
ResidualUNet_256
UNet_NoSkip_256
UNet_Standard_256
0.8
0.6
0.4
0.2
epoch
10
20
30
40
50

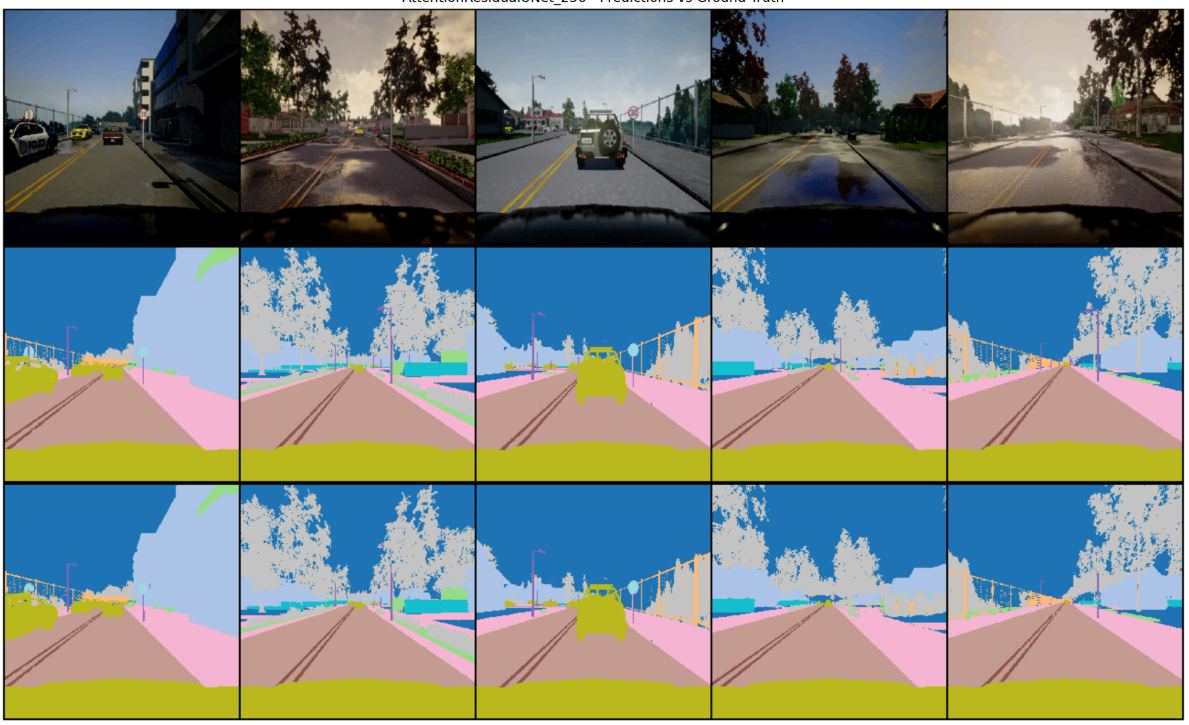

test/miou
UNet_NoSkip_256_test
AttentionResidualUNet_256_test
ResidualUNet_256_test
UNet_Standard_256_test
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9


test/loss
UNet_NoSkip_256_test
AttentionResidualUNet_256_test
ResidualUNet_256_test
UNet_Standard_256_test
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45

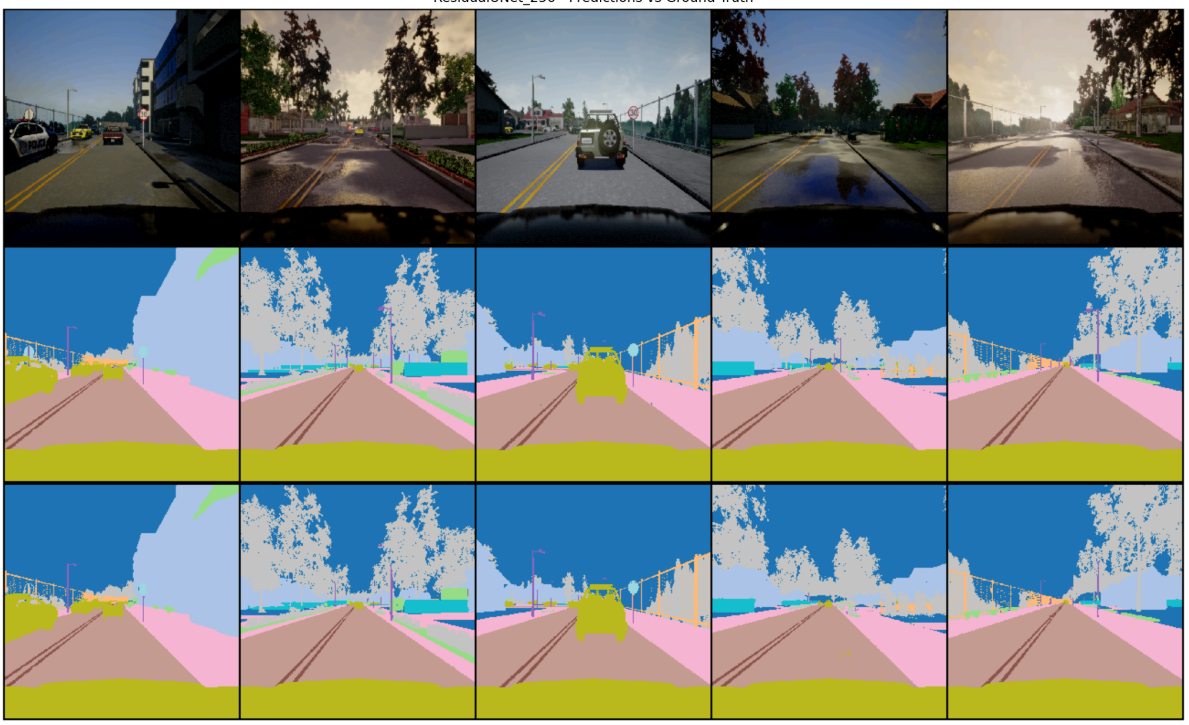


Qualitative Results

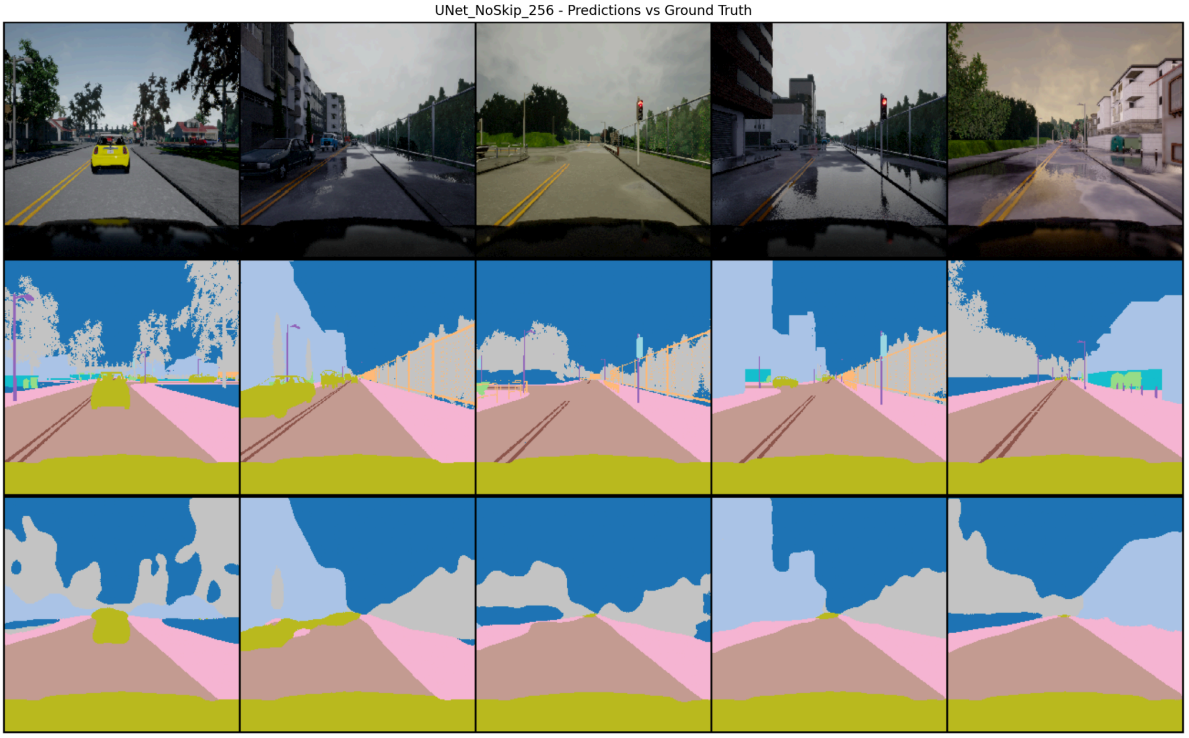
AttentionResidualUNet_256 - Predictions vs Ground Truth



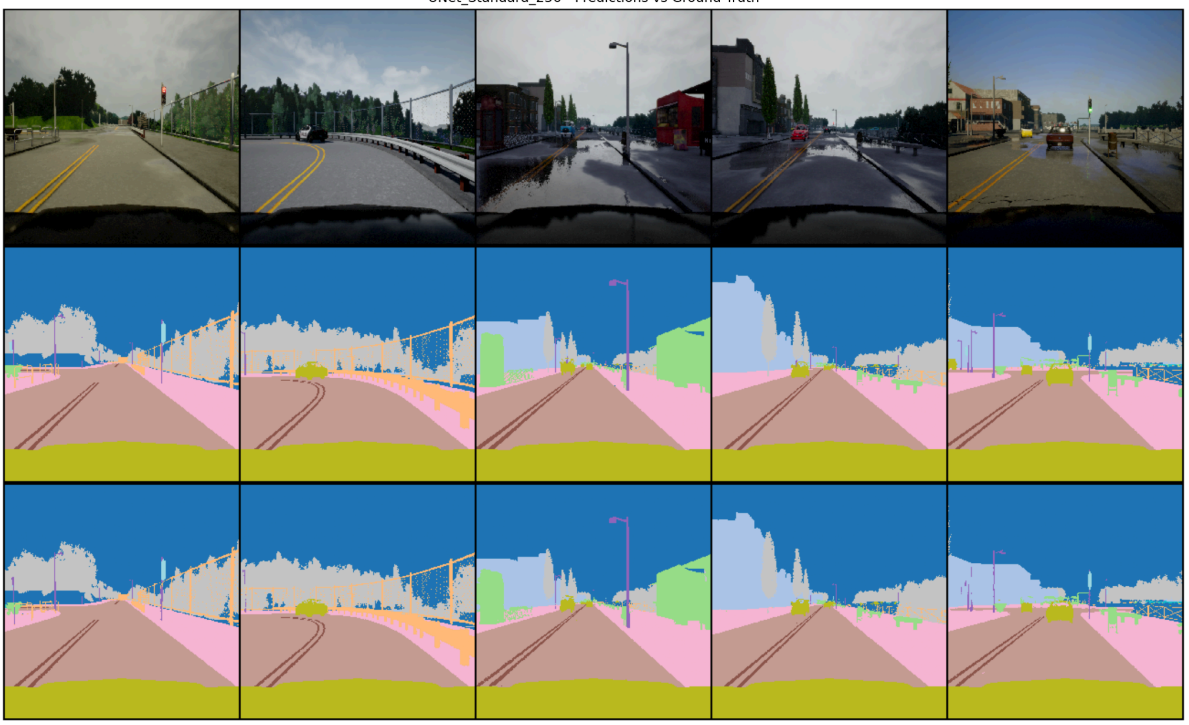
ResidualUNet_256 - Predictions vs Ground Truth



UNet_NoSkip_256 - Predictions vs Ground Truth



UNet_Standard_256 - Predictions vs Ground Truth



Comparison Without Skip Connections

1. Segmentation Quality

•Detail Loss:

The Feature U-Net without skip connections tends to produce blurry and less detailed segmentation maps. On the other hand, the standard U-Net—with skip connections—preserves much sharper details and maintains the structure of objects more effectively.

•Edge Clarity:

Without skip connections, boundaries between objects are often soft or missing altogether. The standard U-Net handles edges much better, clearly separating different regions and retaining fine structural boundaries.

•Small Object Detection:

Skip-less models often miss small objects or localize them poorly. The standard U-Net captures and segments these small features more accurately thanks to the preserved high-resolution features.

•Alignment with Input:

Predictions from the skip-less model don't align well with the shapes in the input image. In contrast, the standard U-Net's outputs closely follow the contours of the input, ensuring much better structural fidelity.

———

2. Why Skip Connections Matter

Skip connections are fundamental to U-Net's architecture and are key to its success in tasks that require pixel-level precision and structure-aware segmentation. Here's why they're so important:

•Preserving Spatial Detail:

As the image passes through the encoder, spatial resolution drops and details get lost. Skip connections bring back these high-res features from the encoder and inject them into the decoder, helping recover lost information.

•Accurate Localization:

U-Net needs to combine deep semantic understanding with fine localization. Skip connections bridge these two by merging low-level spatial cues with high-level features, making the model better at pinpointing where things are.

•Smoother Gradient Flow:

They also help training by improving gradient flow and reducing the chance of vanishing gradients. While this isn't a huge concern in shallow networks like our 4-layer U-Net, it still aids convergence.

•Capturing Small Features:

Small or thin objects often disappear in the deeper layers. Skip connections reintroduce these at the decoding stage, allowing the model to segment them correctly.

•Cleaner Outputs:

Without skip connections, the model relies only on coarse, compressed information, leading to blurry or misaligned segmentations. By restoring high-frequency detail, skip connections produce sharper, more coherent results.

Bottom Line:

Skip connections are what give U-Net its edge—they allow it to balance global context with local detail and generate accurate, high-resolution segmentation maps.

———

Comparison with Gated Attention

1. Why Use Attention (Based on the Paper)

Attention Gates (AGs), introduced in Attention U-Net (Oktay et al., 2018), enhance segmentation in several ways, particularly when images are complex or noisy:

•Focuses on What Matters:

AGs learn to ignore irrelevant parts of the image and concentrate on regions important to the task, which is valuable when only certain parts of the image need segmentation.

•Lightweight Design:

AGs add only a small computational overhead and don't require extra labels or complex training procedures. They integrate seamlessly into the U-Net architecture.

•No Need for ROI Heuristics:

Traditional methods often use region proposals or pre/post-processing to isolate areas of interest. AGs learn this attention mechanism automatically during training.

•Better Generalization:

By filtering out noise and focusing only on meaningful areas, models with AGs often generalize better, especially in the presence of background clutter or variation in object appearance.

Role of the Gating Signal:

The attention gate uses a gating signal from the decoder (which captures high-level context) to filter encoder features before they are passed through the skip connections. This process:

•Adds Contextual Awareness:
The model decides which encoder features are useful based on decoder context—this helps filter out noise and focus on relevant regions.

•Propagates Only Key Features:
Instead of passing everything through the skip connection, AGs selectively forward the important parts, improving object boundary clarity.

•Combines Detail and Context:
AGs ensure that the local details passed through the skip connections are contextually relevant, resulting in sharper and more meaningful segmentations.

———

2. Results in Our Use Case

While AGs are beneficial in some scenarios, in our setup (multi-class, full-image segmentation), there was no significant improvements over standard U-Net. Here's why:

•No Need to "Focus":
AGs work best when only part of the image matters, such as tumor detection. But in our case, every pixel is part of a class, so there's no "irrelevant" region to filter out.

•Dense Segmentation Already Preserves Detail:
Since the task is dense, the model is already learning to keep detail across the entire image. Skip connections alone are effective at retaining both low- and high-level features.

•Low Background Noise:
AGs shine when clutter or occlusion causes confusion. But in our dataset, even smaller cluttered objects are labeled and need to be segmented, so there's little "useless" background to ignore.

•Too Many Classes for Focused Attention:
In binary segmentation, attention can zero in on a single class. But in multi-class segmentation, attention gets spread thin—no single region stands out, so AGs don't get to focus effectively.

•Potential for Improvement with Class-wise AGs:
If we had class-specific attention (e.g., one gate per class), we might see bigger performance gains, especially in distinguishing overlapping or occluded classes.