Comparative Analysis of U-Net Variants for Semantic Segmentation

Krishna Konda (knkonda2@illinois.edu)
Himavanth Mahesh (hmahesh2@illinois.edu)
Minkyung Chung (mc43@illinois.edu)

Abstract:

This project presents a comparative analysis of three U-Net variants, Standard U-Net, Attention U-Net (SCSE), and U-Net++, to investigate the impact of more complex architecture with skip connections on semantic segmentation performance in complex urban driving scenes. By training all models on the same dataset and evaluating performance using mean Intersection over Union (mIoU) and class-wise IoU metrics, we assess the relative benefits of incorporating attention and nested and dense skip connections. We further experiment with three different encoders, ResNet-34, ResNet-50, and MobileNetV2 on Attention U-Net to evaluate how encoder capacity affects downstream performance. Our results show that U-Net++ achieves the highest mIoU as well as the best class-wise IoU scores for categories susceptible to occlusion such as "person" and "car", with Attention U-Net performing second best. These findings highlight how architectural choices in the skip connections significantly influence segmentation quality, particularly for occluded objects.

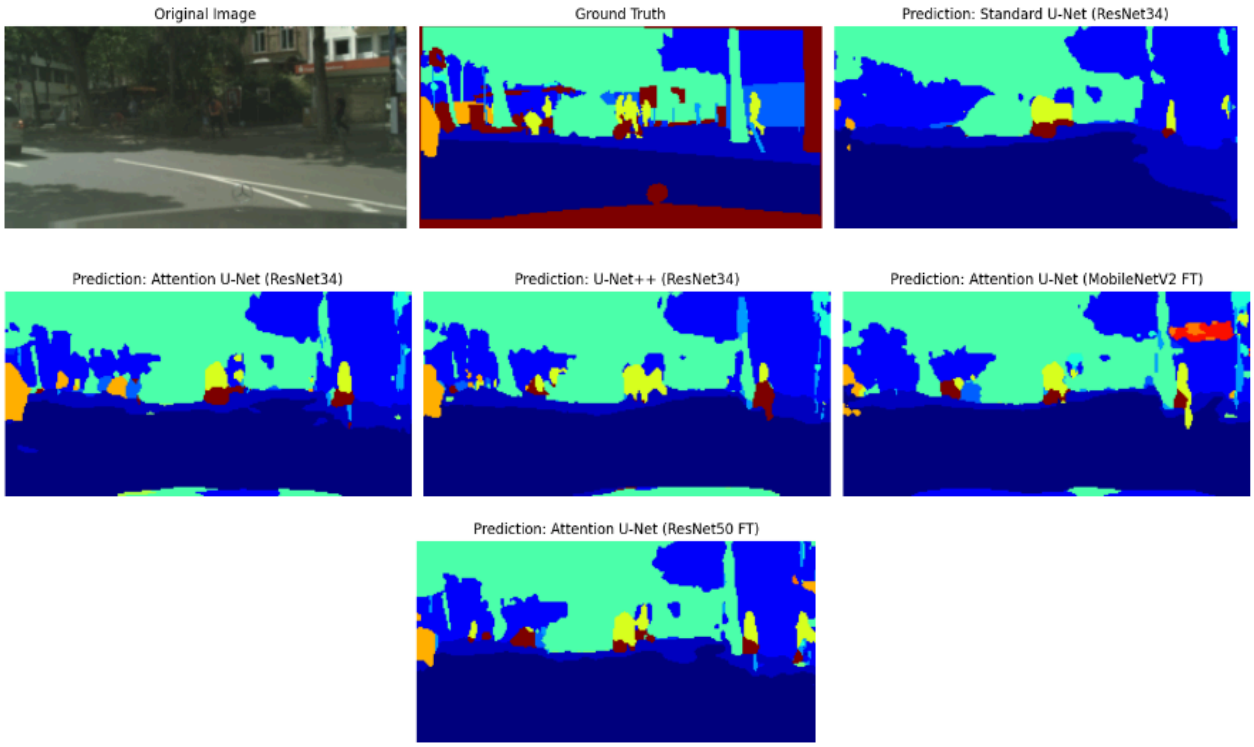


Figure 1: Comparative Analysis of Segmentation across models

All source code used in this project is available at: U-Net Segmentation Project

Introduction

Semantic segmentation is a critical task in computer vision, especially in the domain of autonomous driving, where it enables systems to understand and interpret complex urban environments by assigning a semantic label to each pixel in an image. Accurate segmentation is essential for identifying road elements such as vehicles, pedestrians, traffic signs, and lane markings. However, challenges such as occlusion and fine-grained boundaries often hinder model performance, particularly for small or overlapping objects.

U-Net's encoder-decoder structure effectively preserves detailed information that could be lost from encoding and then decoding through skip connections. These connections directly link feature maps from the encoder's downsampling path to the corresponding layers in the decoder's upsampling path, allowing the decoder to access spatial details learned in the earlier stages.

Building upon this foundation, extensions like Attention U-Net have emerged. Attention U-Net incorporates attention mechanisms within these skip connections. An attention mechanism allows the network to weigh the importance of different feature channels, enabling it to focus on the most relevant information for the segmentation task. This process includes refining the features passed along the skip connections. Another extension we explore is U-Net++, which introduces nested and dense skip connections. Instead of direct connections, U-Net++ features a series of interconnected networks. These nested connections are combined to create denser skip pathways, enabling more complex information sharing between the encoder and decoder.

This project evaluates and compares three U-Net Based architectures above, Standard U-Net, Attention U-Net (SCSE), and U-Net++ with a focus on skipping connection designs and their impact on performance on segmentation in urban driving scenarios. The experiments are conducted using the Cityscapes dataset, which includes approximately 3000 finely annotated images of real-world street scenes.

To further explore the influence of encoder design, we experiment with multiple encoders: ResNet-34, ResNet-50, and MobileNetV2. Prior studies, including the work by Aboussaleh et al. (2023), have shown that the choice of encoder significantly affects both feature extraction quality and downstream segmentation accuracy. Deeper encoders can capture more abstract representations, which may improve performance on complex scenes, while lightweight encoders like MobileNetV2 offer faster inference and lower memory consumption, useful in real-time systems. By varying the encoder, we aim to understand the trade-offs between depth, efficiency, and segmentation accuracy.

Our objective is to identify which architectural components most effectively improve segmentation performance, particularly for occluded or difficult classes such as vehicles and pedestrians. Performance is evaluated using metrics including mean Intersection over Union (mIoU) and class-wise IoU and qualitative analysis on sample images with occlusion objects. Through this comparative study, we aim to gain insights into model design choices that support more accurate and reliable perception systems for autonomous vehicles.

Details of the approach

Our approach involves a comparative evaluation of three U-Net variants—Standard U-Net, Attention U-Net with Spatial and Channel Squeeze-and-Excitation (SCSE) attention, and U-Net++—for semantic segmentation on the Cityscapes dataset. Each model is implemented using the segmentation_models_pytorch library, with ResNet-34 as the default encoder, and we extend the analysis by experimenting with MobileNetV2 and ResNet-50 encoders for Attention U-Net. Below, we detail the dataset preprocessing, model architectures, training pipeline, and evaluation metrics, supported by pseudocode and diagrams to illustrate the workflow.

Dataset Preprocessing

The Cityscapes dataset, comprising 2975 training and 500 validation images with fine annotations for 19 semantic classes, is used for training and evaluation. Images are stored in leftImg8bit_trainvaltest.zip, and annotations in gtFine_trainvaltest.zip, both accessed via Google Drive in a Google Colab environment. To manage computational constraints, images are resized from their original 1024x2048 resolution to 128x256 pixels, and annotations are resized using nearest-neighbor interpolation to preserve label integrity. Labels are mapped from Cityscapes IDs to a 0-18 range for the 19 classes, with unmapped labels set to an ignore_index of 255 to exclude them from loss calculations. The preprocessing steps are implemented as follows:

```python
# Define transforms (resize to manage memory)
# Original image size for Cityscapes is 1024x2048. Resizing significantly.
img_height, img_width = 128, 256 # Reduced size for faster training and lower memory

transform = transforms.Compose([
    transforms.Resize((img_height, img_width)),
    transforms.ToTensor()
])
```

```python
def target_to_tensor(target):
    # Resize target segmentation map. PIL Image object is expected by Resize.
    target = target.resize((img_width, img_height), resample=Image.NEAREST) # Nearest neighbor for labels
    target_np = np.array(target, dtype=np.uint8)

    # Map Cityscapes labels to 0-18 (for 19 classes), set others to ignore_index (255)
    label_map = {
        # name: id, trainId
        'unlabeled': (0, 255), 'ego vehicle': (1, 255), 'rectification border': (2, 255),
        'out of roi': (3, 255), 'static': (4, 255), 'dynamic': (5, 255),
        'ground': (6, 255), 'road': (7, 0), 'sidewalk': (8, 1),
        'parking': (9, 255), 'rail track': (10, 255), 'building': (11, 2),
        'wall': (12, 3), 'fence': (13, 4), 'guard rail': (14, 255),
        'bridge': (15, 255), 'tunnel': (16, 255), 'pole': (17, 5),
        'polegroup': (18, 255), 'traffic light': (19, 6), 'traffic sign': (20, 7),
        'vegetation': (21, 8), 'terrain': (22, 9), 'sky': (23, 10),
        'person': (24, 11), 'rider': (25, 12), 'car': (26, 13),
        'truck': (27, 14), 'bus': (28, 15), 'caravan': (29, 255),
        'trailer': (30, 255), 'train': (31, 16), 'motorcycle': (32, 17),
        'bicycle': (33, 18)
    }
    mapped_target = np.full_like(target_np, 255, dtype=np.uint8) # 255 for ignore_index

    for cityscapes_id_tuple, train_id in label_map.items():
        original_id = train_id[0]
        target_train_id = train_id[1]
        if target_train_id != 255:
            mapped_target[target_np == original_id] = target_train_id

    return torch.from_numpy(mapped_target).long()
```

Model Architectures

All models use an encoder-decoder structure, with variations in skip connection mechanisms:

- **Standard U-Net**: Features direct skip connections between corresponding encoder and decoder layers, concatenating feature maps to preserve spatial details during upsampling.
- **Attention U-Net (SCSE)**: Incorporates SCSE attention modules in the decoder, which reweight feature channels and spatial regions to emphasize relevant features, particularly for occluded objects.
- **U-Net++**: Uses nested and dense skip connections, creating intermediate convolutional layers between encoder and decoder paths to enhance feature aggregation.

Each model uses a ResNet-34 encoder pre-trained on ImageNet, except for additional experiments with Attention U-Net using MobileNetV2 (lightweight, for efficiency) and ResNet-50 (deeper, for richer features). The models are configured with 3 input channels (RGB) and 19 output classes, outputting raw logits for compatibility with CrossEntropyLoss.

```python
# Step 7: Define Models
import segmentation_models_pytorch as smp

num_classes = 19  # Based on the label mapping (0-18)
encoder_name = "resnet34"

# Standard U-Net
model_standard = smp.Unet(
    encoder_name=encoder_name,
    encoder_weights="imagenet", # Using pre-trained weights can help convergence
    in_channels=3,
    classes=num_classes,
    activation=None # Raw logits for CrossEntropyLoss
)
print(f"Standard U-Net with {encoder_name} encoder defined.")

# Attention U-Net (with SCSE attention)
model_attention = smp.Unet(
    encoder_name=encoder_name,
    encoder_weights="imagenet",
    in_channels=3,
    classes=num_classes,
    activation=None,
    decoder_attention_type="scse" # Spatial and Channel Squeeze & Excitation
)
print(f"Attention U-Net (SCSE) with {encoder_name} encoder defined.")

# U-Net++
model_unetpp = smp.UnetPlusPlus(
    encoder_name=encoder_name,
    encoder_weights="imagenet",
    in_channels=3,
    classes=num_classes,
    activation=None
)
print(f"U-Net++ with {encoder_name} encoder defined.")
```

Training Pipeline

Training was conducted in Google Colab with GPU acceleration, using the Adam optimizer (learning rate 0.001, weight decay 1e-5) and CrossEntropyLoss with ignore_index=255. Models are trained for 15 epochs, followed by 5 epochs of fine-tuning for Attention U-Net and U-Net++ with a reduced learning rate of 5e-5. A ReduceLROnPlateau scheduler adjusts the learning rate based on validation loss. The training loop includes:

1. **Forward Pass**: Input images are passed through the model to compute logits.
2. **Loss Computation**: CrossEntropyLoss is calculated between logits and target labels.
3. **Backward Pass**: Gradients are computed and parameters updated via Adam.
4. **Validation**: After each epoch, validation loss is computed, and the best model (lowest validation loss) is saved.
5. **Fine-Tuning**: Pre-trained weights are loaded, and training continues with a lower learning rate to refine weights.

Evaluation Metrics

Performance is evaluated using mean Intersection over Union (mIoU) and class-specific IoU for "person" (class 11) and "car" (class 13), which are prone to occlusion. For each batch, predictions are obtained via argmax on model logits, and IoU is calculated per class, with NaN values (for absent classes) excluded from the mean. The evaluation loop processes the validation set, aggregating mIoU and class IoUs.

```
# Step 10: Evaluation Function for IoU
def compute_iou(outputs, targets, num_classes, ignore_index=255):
    outputs = torch.argmax(outputs, dim=1) # Get predicted class for each pixel

    outputs = outputs.cpu()
    targets = targets.cpu()

    iou_per_class = []
    for cls in range(num_classes): # Iterate from 0 to num_classes-1
        pred_inds = (outputs == cls)
        target_inds = (targets == cls)

        intersection = (pred_inds & target_inds).sum().item()
        union = (pred_inds | target_inds).sum().item()

        if union == 0:
            iou_per_class.append(float('nan'))
        else:
            iou_per_class.append(intersection / union)

    # Calculate mean IoU, ignoring NaN values (classes not present in this batch)
    valid_iou = [iou for iou in iou_per_class if not np.isnan(iou)]
    mean_iou_batch = np.nanmean(valid_iou) if len(valid_iou) > 0 else float('nan')

    return mean_iou_batch, iou_per_class
```

Visualization

To qualitatively assess performance, predictions are visualized for a validation image containing "person" or "car" objects. The original image, ground truth, and model predictions are displayed side-by-side using matplotlib, with a jet colormap for segmentation masks. This step highlights differences in handling occluded objects across models.

Intermediate Stages

Intermediate results include training and validation losses per epoch, saved model checkpoints, and IoU metrics. For example, during training, we monitor loss convergence and validate model performance on a subset of the validation set.

Diagram

A conceptual diagram of the workflow includes:

- **Input**: Cityscapes images and annotations.

- **Preprocessing**: Resizing and label mapping.
- **Models**: U-Net variants with different skip connections and encoders.
- **Training**: Epoch-wise optimization with loss computation.
- **Evaluation**: IoU metrics and visualization of predictions.

This structured approach ensures a thorough comparison of U-Net variants, focusing on architectural impacts on segmentation performance in urban scenes.

Results

This section presents the quantitative and qualitative outcomes of our comparative analysis of U-Net variants and encoder configurations on the Cityscapes validation set. All models were trained for 15 epochs, with the U-Net variants (Standard U-Net, Attention U-Net, and U-Net++) using a ResNet-34 encoder also undergoing an additional 5 epochs of fine-tuning with a reduced learning rate. The Attention U-Net architecture was further evaluated with MobileNetV2 and ResNet50 encoders, which were also fine-tuned. Performance was primarily assessed using mean Intersection over Union (mIoU) and class-specific IoU for "person" and "car" classes.

Quantitative Performance Metrics

The performance metrics for each model configuration are summarized in Table 1. These results reflect the performance of the best model checkpoint saved during training (or fine-tuning) based on validation loss.

Table 1: Semantic Segmentation Performance on Cityscapes Validation Set

Model Configuration	Encoder	Mean IoU (mIoU)	Person IoU (Class 11)	Car IoU (Class 13)
Standard U-Net (Fine-tuned)	ResNet-34	0.3165	0.2180	0.6441
Attention U-Net (SCSE) (Fine-tuned)	ResNet-34	0.3281	0.2425	0.6984
U-Net++ (Fine-tuned)	**ResNet-34**	**0.3408**	0.2578	**0.7235**
Attention U-Net (SCSE) (MobileNetV2 FT)	MobileNetV2	0.3350	**0.2675**	0.6989
Attention U-Net (SCSE) (ResNet50 FT)	ResNet-50	0.3378	0.2260	0.6919

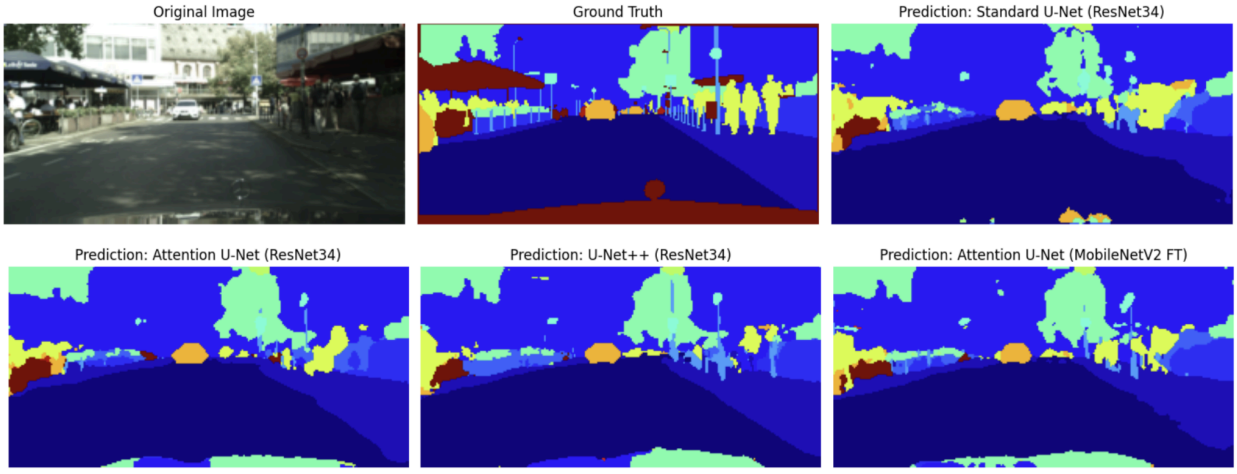
Note: "FT" denotes fine-tuned models. The highest scores for each metric among all models are in bold

Key Observations from Quantitative Results:

1. **Comparison of U-Net Variants (ResNet-34 Encoder):**
    - Among the models using the ResNet-34 encoder, U-Net++ (Fine-tuned) demonstrated the highest overall performance, achieving an mIoU of 0.3408. It also yielded the highest Car IoU (0.7235) and a competitive Person IoU (0.2578).
    - Attention U-Net (ResNet-34 FT) showed improved performance over the Standard U-Net (ResNet-34 FT), with mIoUs of 0.3281 and 0.3165, respectively. This trend was also observed for both Person and Car IoUs.
2. **Impact of Encoder on Attention U-Net Performance:**
    - When evaluating the Attention U-Net (SCSE) architecture with different encoders, the ResNet-50 encoder achieved the highest mIoU (0.3378) among these specific Attention U-Net configurations.
    - The Attention U-Net with a MobileNetV2 encoder (FT) yielded a strong mIoU of 0.3350, slightly outperforming the ResNet-34 version (0.3281) in mIoU.
    - Notably, the Attention U-Net (MobileNetV2 FT) achieved the highest Person IoU across all tested models at 0.2675.
    - For Car IoU, the Attention U-Net variants with ResNet-34, MobileNetV2, and ResNet-50 encoders all performed similarly (0.6984, 0.6989, and 0.6919 respectively), with these scores being generally higher than the Standard U-Net but lower than U-Net++ with ResNet-34.
3. **Overall Best Performers:**
    - U-Net++ (ResNet-34 FT) achieved the highest mIoU (0.3408) and the highest Car IoU (0.7235) overall.
    - Attention U-Net (MobileNetV2 FT) achieved the highest Person IoU (0.2675) overall.

Qualitative Results

Visual inspection of segmentation outputs on a representative validation image from the Cityscapes dataset complements the quantitative metrics. This image displays a side-by-side comparison of the original input image, the ground truth segmentation, and the predictions from the five evaluated model configurations: Standard U-Net (ResNet-34 FT), Attention U-Net (ResNet-34 FT), U-Net++ (ResNet-34 FT), Attention U-Net (MobileNetV2 FT), and Attention U-Net (ResNet50 FT).



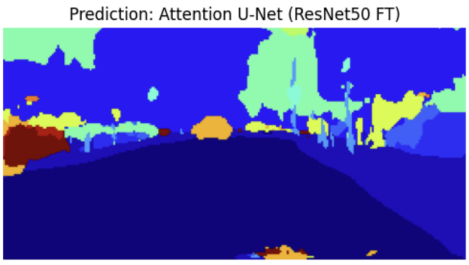


Figure 2: Comparative Analysis of Segmentation across models (1)

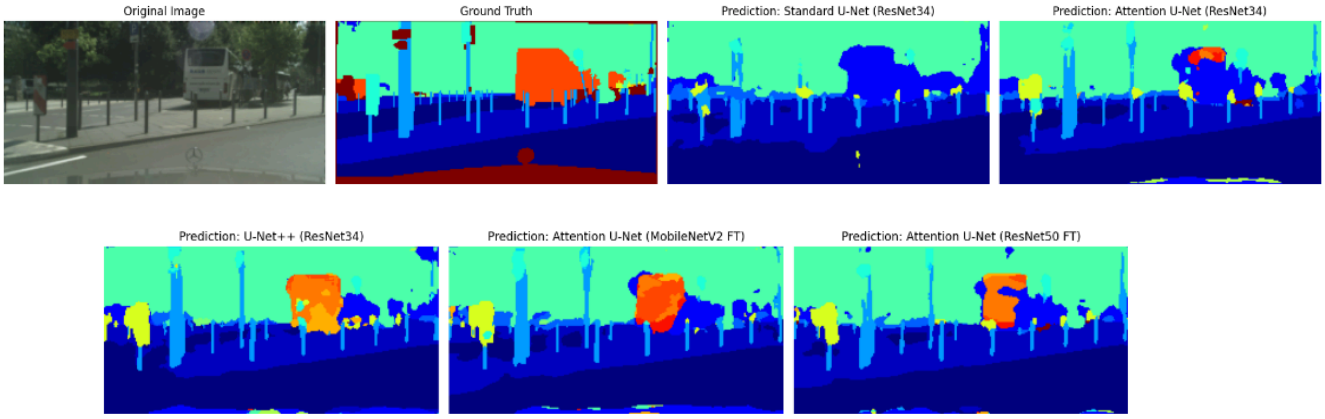


Figure 3: Comparative Analysis of Segmentation across models (2)

A critical factor influencing the qualitative performance, particularly for smaller object classes, is the significant downscaling of images from their original 1024x2048 resolution to 128x256 pixels for training and inference. This reduction was necessary for managing computational resources but inherently leads to a loss of fine-grained detail. This is especially pertinent for classes like "person," where individuals often occupy a small pixel area in the original image and become even more challenging to discern and accurately segment after aggressive resizing.

Observations from Image Comparison

- **General Scene Understanding:** All models successfully capture the macro-level structure of the urban scene, including the general layout of the road, buildings, and sky. The dominant classes are recognized by all architectures.

- **Impact of Image Resizing on "Person" Class:** The ground truth reveals a very small instance of a person on the far-left sidewalk. As anticipated, due to the extreme image downscaling, this small object proves exceptionally challenging for all models. In the provided visual, distinct and accurate segmentation of this specific person instance is not clearly evident in any of the model predictions, underscoring the difficulty posed by the low resolution for such small-scale objects. The reported quantitative IoU scores for the "person" class, while showing relative differences (with Attention U-Net MobileNetV2 FT scoring highest), are generally modest, which is consistent with this visual difficulty.

- **Segmentation of "Car" Class:**

    - Vehicles, being larger objects, are more consistently segmented across all models.

- The **U-Net++ (ResNet34 FT)** prediction appears to offer the most coherent and well-defined segmentation for the cars, with relatively clean boundaries and less fragmentation. This visual observation aligns with its highest reported Car IoU (0.7235).
- The **Attention U-Net variants (ResNet-34 FT, MobileNetV2 FT, and ResNet50 FT)** also produce strong segmentations for cars, visually appearing quite similar to each other in this example and generally cleaner than the Standard U-Net. Their Car IoU scores were quantitatively similar and higher than Standard U-Net.
- **Standard U-Net (ResNet34 FT)**, while correctly identifying car regions, tends to produce slightly more blocky or less refined boundaries for vehicles compared to the other, more complex architectures.

- **Occluded Object Segmentation:**

  - A key example of occlusion appears in Figure 3, where a large bus is partially obscured by surrounding objects. This instance reveals notable differences across architectures. Standard U-Net (ResNet34 FT) fails to detect the bus entirely, showing no segmentation response in the occluded region. In contrast, Attention U-Net models begin to register the presence of the bus, with faint red markings indicating partial detection. The strongest and most coherent segmentation of the bus is produced by U-Net++ (ResNet34 FT), which captures both the shape and approximate boundaries despite the occlusion. This is the most accurate result among models using the ResNet34 encoder. Models with other encoders, such as MobileNetV2, show even stronger responses.

- **Comparison of U-Net Variants (ResNet-34 Encoder):**

  - Visually, both Attention U-Net and U-Net++ with the ResNet-34 encoder show an improvement in segmentation quality over the Standard U-Net, particularly in terms of object cohesion and boundary definition for the larger car objects. U-Net++ appears to have a slight edge in visual clarity in this specific example.

- **Impact of Encoders on Attention U-Net:**

  - All three Attention U-Net variants (with ResNet-34 FT, MobileNetV2 FT, and ResNet50 FT encoders) provide visually competent segmentations of the scene.
  - The **Attention U-Net (MobileNetV2 FT)**, despite using a lightweight encoder, produces a segmentation map that is visually comparable to those from the ResNet-based encoders for major structures and cars in this image. While its superior quantitative performance on the "person" class (0.2675 IoU) is difficult to definitively confirm from this single visual due to the object's small size and the image resolution, its overall visual output does not suggest a significant compromise in this scene compared to the deeper encoders.
  - The **Attention U-Net (ResNet50 FT)** also delivers a clean segmentation, visually similar to the ResNet-34 Attention U-Net for the prominent objects.

In summary, the qualitative results highlight that while all models grasp the general scene context, the architectural enhancements in U-Net++ and Attention U-Net, particularly with capable encoders, tend to yield more refined segmentations for larger, more distinct objects like cars. The severe image downscaling poses a significant challenge for all models in accurately segmenting very small objects like distant pedestrians, which is an important consideration when interpreting both qualitative and quantitative results for such classes.

Discussion

Our comparative analysis of U-Net variants (Standard U-Net, Attention U-Net with SCSE, and U-Net++) and different encoder backbones (ResNet-34, ResNet-50, MobileNetV2) on the Cityscapes dataset has yielded several key insights into their semantic segmentation capabilities, particularly within the constraints of a significantly reduced image resolution (128x256).

Impact of Architectural Design (Skip Connections): The results (Table 1) indicate a clear trend: more sophisticated skip connection mechanisms generally lead to improved segmentation performance when using the same ResNet-34 encoder. U-Net++ (ResNet-34 FT), with its nested and dense skip pathways, achieved the highest overall mIoU (0.3408) and the best Car IoU (0.7235). This suggests that enabling richer feature fusion across multiple scales within the decoder, as U-Net++ does, is beneficial for capturing complex contextual information and refining object boundaries, especially for larger and more prominent classes like cars. The qualitative results visually support this, showing U-Net++ providing more coherent segmentations for vehicles.

Attention U-Net (SCSE, ResNet-34 FT) also outperformed the Standard U-Net (ResNet-34 FT) in terms of mIoU (0.3281 vs. 0.3165) and for both "person" and "car" classes. This aligns with the expectation that attention mechanisms, like SCSE which recalibrates channel-wise and spatial features, can help the model focus on more salient information within the feature maps passed through skip connections, leading to better discrimination and localization.

Impact of Encoder Choice for Attention U-Net: The experiments with different encoders for the Attention U-Net architecture revealed interesting trade-offs. As anticipated, the deeper ResNet-50 encoder (Attention U-Net ResNet50 FT) yielded a slightly higher mIoU (0.3378) compared to the ResNet-34 (0.3281) and MobileNetV2 (0.3350) encoders with the same Attention U-Net setup. This suggests that increased model capacity in the encoder can indeed capture more complex feature representations beneficial for overall segmentation.

However, a particularly noteworthy and somewhat surprising finding was the performance of Attention U-Net with the MobileNetV2 encoder. Despite being a lightweight encoder designed for efficiency, it achieved a very competitive mIoU (0.3350) and, most strikingly, the highest "person" IoU (0.2675) across all tested models, surpassing even U-Net++ (ResNet-34 FT) on this specific metric. This outcome suggests that for smaller, harder-to-detect objects like persons, especially at very low resolutions, the feature characteristics extracted by MobileNetV2, when combined with an attention mechanism, might be particularly effective. It's possible that the less complex feature set of MobileNetV2 prevents over-suppression of subtle cues from small objects, which deeper networks might smooth over or lose amidst more dominant features, especially given the aggressive downsampling. This finding highlights that a deeper or more complex encoder is not universally superior across all object classes or conditions, and efficiency-focused encoders can still yield strong results for specific challenging sub-tasks.

Performance on Challenging Classes ("Person" and "Car"): Our focus on "person" and "car" classes, often subject to occlusion, revealed that U-Net++ (ResNet-34 FT) excelled at segmenting cars, likely due to its enhanced ability to integrate multi-scale features which helps in delineating larger, structured objects. The superior performance of Attention U-Net (MobileNetV2 FT) on the "person" class is a key finding, suggesting its architecture might be more adept at localizing smaller, less distinct objects under severe resolution constraints. The qualitative analysis visually corroborates the difficulty in segmenting persons due to downscaling, making the quantitative superiority of the MobileNetV2 variant for this class even more interesting.

Limitations of the Study: It is crucial to acknowledge the limitations of this study, which temper the generalizability of the mIoU scores achieved:

1. **Image Resolution:** The most significant constraint was the use of a very low image resolution (128x256) compared to Cityscapes' native resolution. This dramatically reduces the available detail, making segmentation of small objects (like persons, distant cars, poles) and fine boundaries inherently difficult for any model. The reported mIoU values are therefore specific to this low-resolution setting and not directly comparable to state-of-the-art benchmarks typically reported at much higher resolutions.
2. **Training Duration and Computational Resources:** The models were trained for 15 epochs with 5 epochs of fine-tuning. While convergence was observed in validation loss, more extended training might yield further improvements or alter the relative performance, especially if learning rates were allowed to decay further. Our experiments were conducted on Google Colab, which has time and resource limitations.
3. **Scope of Encoder Variation:** Encoder variations (ResNet-50, MobileNetV2) were only applied to the Attention U-Net architecture. Applying these to Standard U-Net and U-Net++ could provide a more complete picture of the encoder-architecture interplay.
4. **Data Augmentation:** Only basic resizing and ToTensor transforms were used. More extensive data augmentation techniques could improve model robustness and generalization, potentially leading to higher IoU scores.

Conclusion

This comparative study on U-Net variants for semantic segmentation on the Cityscapes dataset, despite operating under significant resolution constraints, provides valuable insights. Our findings indicate that:

1. Architectural enhancements to skip connections significantly impact performance. U-Net++, with its nested and dense skip connections, demonstrated the best overall mIoU and superior performance in segmenting larger objects like cars when using a ResNet-34 encoder.
2. Attention mechanisms (specifically SCSE in Attention U-Net) offer an improvement over the Standard U-Net, suggesting that guided feature fusion is beneficial.
3. The choice of encoder plays a complex role. While a deeper encoder (ResNet-50) provided a slight mIoU edge for Attention U-Net, the lightweight MobileNetV2 encoder surprisingly achieved the best performance for the challenging "person" class, highlighting that model capacity needs to be considered in conjunction with object scale and image resolution.
4. The severe image downscaling was a major limiting factor, particularly for small object classes like "person," emphasizing the importance of input resolution for achieving high-fidelity segmentation in complex urban scenes.

Future work should prioritize experiments at higher image resolutions to better align with real-world application requirements and allow for more direct comparison with established benchmarks. Exploring a wider range of data augmentation techniques, more extensive training, and applying encoder variations across all U-Net architectures would further clarify the strengths and weaknesses of each approach. Nevertheless, this study underscores that even with constrained resources, thoughtful comparison of architectural choices can reveal important design considerations for semantic segmentation models. The effectiveness of U-Net++ for general robustness and the surprising efficacy of a MobileNetV2-backed Attention U-Net for small objects provide interesting avenues for developing perception systems optimized for specific aspects of autonomous driving.

Individual Contribution

Krishna Konda (knkonda2): Krishna was responsible for the project scaffolding within Google Colab, including the data pipeline (dataset downloading, preprocessing, and loading). He implemented the shared evaluation utilities, including the mean IoU calculation logic and the code for generating segmentation visualizations. Krishna also took lead on the training and fine-tuning of the Attention U-Net model using the ResNet-34 encoder.
Minkyung Chung (mc43): Minkyung was responsible for the training and fine-tuning of the Standard U-Net model. She also conducted the experiments involving different encoder backbones for the Attention U-Net architecture, specifically training and fine-tuning the Attention U-Net with MobileNetV2 and ResNet-50 encoders.
Himavanth Mahesh (hmahesh2): Himavanth was responsible for implementing, training, and fine-tuning the U-Net++ model. He managed the experimental runs for this architecture and contributed to the analysis of its performance relative to the other U-Net variants.

Reference

Aboussaleh I, Riffi J, Fazazy KE, Mahraz MA, Tairi H. Efficient U-Net Architecture with Multiple Encoders and Attention Mechanism Decoders for Brain Tumor Segmentation. *Diagnostics*. 2023; 13(5):872. https://doi.org/10.3390/diagnostics13050872

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for the Pancreas.

Oktay, O., et al. (2018). Attention U-Net: Learning Where to Look for the Pancreas. Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., & Liang, J. (2018). U-Net++: A Nested U-Net Architecture for Medical Image Segmentation

Cordts, M., et al. (2016). The Cityscapes Dataset for Semantic Urban Scene Understanding.

Segmentation Models PyTorch. GitHub Repository.
[https://github.com/qubvel-org/segmentation_models.pytorch]