Mini-Project 2: Semantic Segmentation with Pascal VOC 2007

Lavinia Xu

November 23, 2025

1 Introduction

Semantic segmentation assigns a class label to every pixel in an image, allowing a model to understand both the objects present and their spatial arrangement. This project focuses on applying three segmentation methods to the `PASCAL VOC 2007` dataset, a widely used benchmark that contains everyday scenes with objects such as people, animals, and vehicles. Each image comes with a pixel-level mask, which provides a strong foundation for training and evaluating segmentation models.

The example below shows the type of output that this project aims to produce. The left side is the original image, and the right side shows the cleaned segmentation mask with color-coded classes.

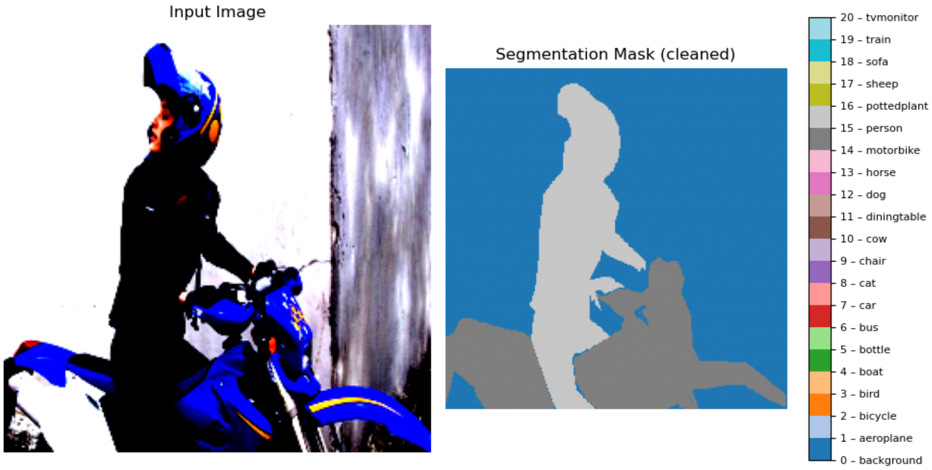

Figure 1: A random sample from dataset and correspinding labels and masks

In this project, I implemented and compared three approaches: a U-Net built from basic convolutional blocks, a DeepLabV3 model using a ResNet backbone, and a foreground-union version of the Segment Anything Model (SAM). These models cover a range of architectures, from fully supervised training to large-scale pretrained systems. The goal of the project is to understand how different design choices influence segmentation outputs and to examine the strengths of each model under the same dataset and evaluation pipeline.

Through these experiments, the project provides a look at how segmentation models operate on `VOC` images and how architectural differences affect their predictions. The main goals of the project are:

- Prepare and normalize the `VOC` dataset for consistent training and evaluation.
- Train three segmentation models under one pipeline.
- Evaluate each model using pixel accuracy, Intersection-over-Union (IoU), Dice score, and HD95.
- Visualize predictions to understand where each model succeeds or struggles.
- Run ablation studies to see how architectural choices or training settings affect performance.

2 Methods

2.1 Data Preprocessing

The dataset already comes with predefined splits. Following the project instructions, I used the training set for model training and treated the validation set as the test set. The original `VOC` test set was not used. Each raw RGB image was resized to a fixed resolution and normalized channel-wise using the ImageNet mean and standard deviation. This step helps stabilize optimization and makes the inputs compatible with models such as DeepLabV3 that were originally trained on ImageNet images. The segmentation masks were resized using nearest-neighbor interpolation to avoid introducing mixed labels.

The mask values use standard `VOC` label IDs where values range from 0 to 20 for the semantic classes and 255 for the regions marked as `void`. During training and evaluation, all loss functions and metric computations ignore pixels labeled 255.

2.2 Model Choices

This project compares three different segmentation approaches:

- **U-Net** is an encoder–decoder architecture with skip connections. The encoder compresses spatial information into deeper feature maps, while the decoder gradually upsamples and reconstructs a full-resolution prediction. Skip connections bring back fine-grained details that helps recover boundaries. U-Net is light and easy to train, making it a good baseline model for this dataset.
- **DeepLabV3** uses atrous (dilated) convolutions and an Atrous Spatial Pyramid Pooling (ASPP) module to capture multi-scale context. It is paired with a pretrained **ResNet-50** backbone to begin with strong feature representations. This allows DeepLabV3 to handle complex object shapes and varied scene structures more better than a basic encoder–decoder model.
- The **Segment Anything Model** (SAM) is a large vision model developed for promptable image segmentation. It combines a powerful image encoder with a mask decoder that can segment objects using points, boxes, or automatically generated mask proposals. In this project, SAM was used in an automatic mode where it produces multiple candidate masks that are combined into a single unified foreground prediction. Its design allows it to generalize to a wide range of visual scenes without additional training.

2.3 Training Procedure

All models were trained on the same normalized training split and evaluated on the same validation split. U-Net and DeepLabV3 were optimized using cross-entropy loss with the `void` index ignored. Training used mini-batches and Adam optimizers where learning rates are tuned individually for the two architectures. SAM does not require training in this setup but still follows the same preprocessing and evaluation steps for a fair comparison.

2.4 Evaluation Metrics

To measure performance consistently across models, the following metrics were used to capture not only how often a model predicts correctly but also how well it aligns object shapes and boundaries:

- **Pixel accuracy** measures the proportion of pixels whose predicted class matches the ground truth, giving an overall sense of labeling correctness across the image.
- **Mean Intersection-over-Union** (mIoU) quantifies the ratio between the overlap and the union of predicted and true regions across all classes.
- **Dice score** evaluates how well the predicted region aligns with the ground truth by emphasizing overlap, which makes it particularly useful for smaller or less frequent objects.
- **HD95** computes the 95th percentile of the Hausdorff distance between prediction and ground-truth boundaries, reflecting how accurately object shapes and edges are captured.

3 Results

3.1 Overall Performance

The three models showed very different levels of performance on the `Pascal VOC 2007` dataset. The table below summarizes a quantitative comparison of the three models, using mIoU and pixel accuracy as the primary evaluation metrics.

Model Type	mIoU	Pixel Accuracy
U-Net	0.0431	0.7365
DeepLabV3	**0.4545**	**0.8794**
SAM	0.1780	0.3559

Across all evaluation metrics, **DeepLabV3** achieved the strongest segmentation performance on the validation set. It obtained the highest mIoU of 0.4545 and pixel accuracy of 0.8794, an indicator of both better object localization and more reliable pixel-level predictions. SAM performed moderately well despite not predicting semantic classes and U-Net produced the lowest mIoU, suggesting difficulty separating most object categories except for a few classes.

3.2 U-Net

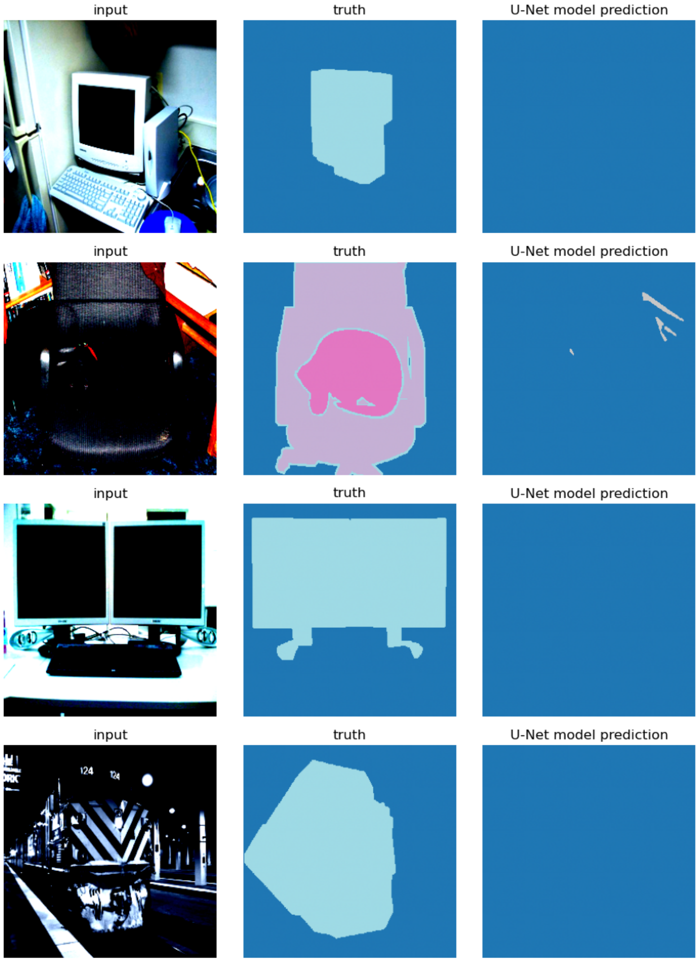

Figure 2: Predictions on validation images from U-Net

U-Net successfully predicted background regions and occasionally recognized people, but it struggled with most other object categories. Its per-class pixel accuracies were near zero for 19 out of 21 classes (`background` included), and boundary alignment remained inconsistent, reflected by a high HD95 value of **86.5352**. The relatively high pixel accuracy are mainly from correctly labeling background pixels which outnumber other more meaningful pixels

Visual inspection of the predictions shown above indicates that U-Net often defaulted to large background segments with fragmented or missing objects. The graph of per-class pixel accuracy shown below confirms a steep imbalance, with `background` far outperforming all other classes.

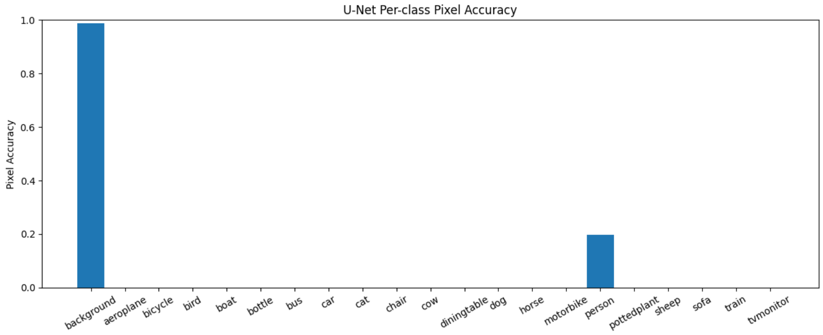

Figure 3: Per-class pixel accuracy for U-Net

3.3 DeepLabV3

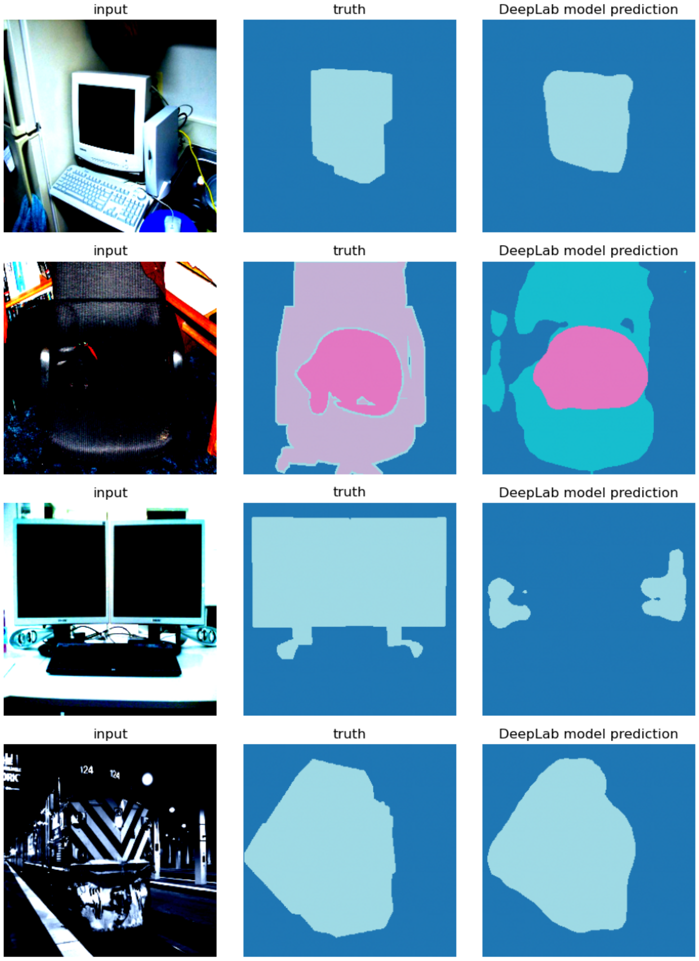

Figure 4: Predictions on validation images from DeepLabV3

DeepLabV3 delivered strong segmentation performance across most categories. It achieved the highest mIoU, Dice score (0.0538), and pixel accuracy, along with a much lower HD95 of **8.4131**, indicating better boundary precision. Many common `VOC` classes, such as person, car, cat, etc., reached meaningful pixel accuracies, while only a few small or infrequent objects remained challenging.

The visual examples in Figure 4 reveal sharper object boundaries and fewer misclassified regions, and the per-class accuracy bar plot below shows more balanced recognition across categories.

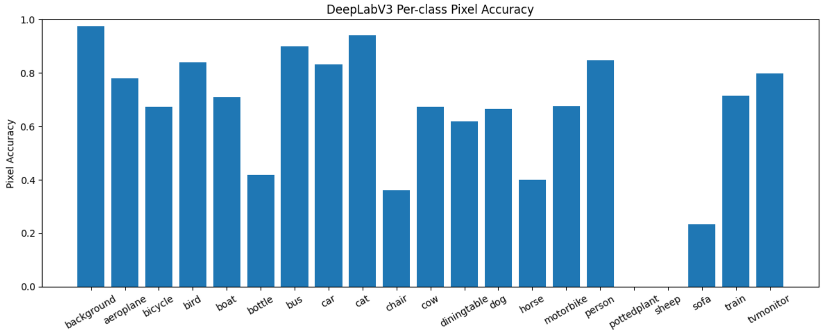

Figure 5: Per-class pixel accuracy for DeepLabV3

3.4 SAM

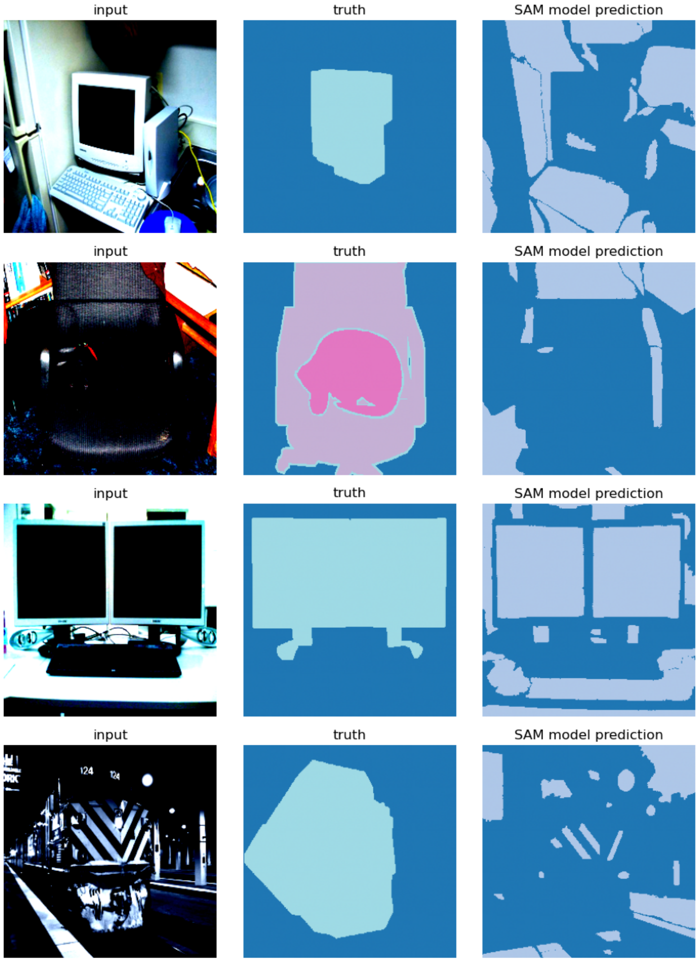

Figure 6: Predictions on validation images from SAM

Since SAM produces a single merged foreground mask rather than 21 semantic labels, its performance naturally differs from the supervised models. It achieved moderate mIoU and Dice scores (0.2625), and showed perfect boundary alignment for many images, which explains the HD95 value of **0.0**. However, pixel accuracy was lower because the model cannot differ between object categories and background as precisely.

Qualitative results in Figure 6 show that SAM often captures the main foreground shape but may under-segment or over-segment depending on scene complexity.

4 Ablation Studies

Ablation studies are an important part of understanding what truly drives segmentation performance. By changing one design choice at a time while keeping everything else identical, we can isolate how much a specific architectural or training decision contributes to the final results. In this project, I conducted three ablation experiments to examine how U-Net model capacity affects learning, whether applying class weighted loss improves U-Net performance, and how pretrained weights influence DeepLabV3. These controlled comparisons reveal trade offs between accuracy, generalization, and computational cost, offering deeper insight into why the models behave the way they do rather than only reporting their final metrics.

4.1 Effect of Channel Size on U-Net

This ablation compares two U-Net configurations that are identical in every way except for their base channel size. The standard model uses 64 channels in its first convolutional block and the smaller version uses 32. Both models were trained on the exact same setup otherwise. The goal was to isolate how much model capacity alone affects segmentation performance. The results are summarized below:

Channel Size	mIoU	Dice Score	Pixel Accuracy	HD95	Runtime
64	0.0431	0.0527	0.7365	78.1221	205 min
32	0.0388	0.0476	0.7278	122.5915	63 min

The results show that decreasing the channel size reduces overall performance. The smaller model reaches worse values across all evaluation metrics. The increase of HD95 suggests its predicted boundaries deviate more from the ground truth. Although both versions classify background well, the 64 channel model handles additional classes slightly better. The bar plot of pixel accuracy shown below highlights this performance gap visually.

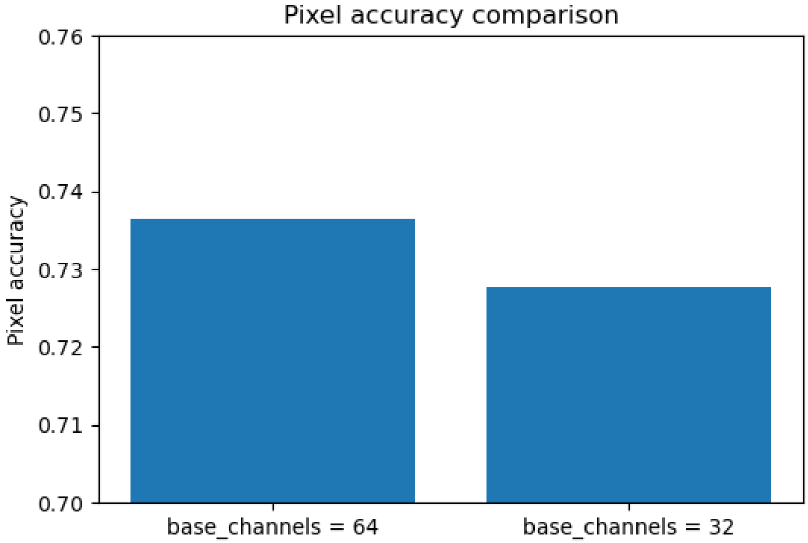

Figure 7: Effect of channel size on U-Net

4.2 Effect of Applying Class Weights on U-Net

This ablation examines whether class weighting helps U-Net handle the strong label imbalance in `Pascal VOC`, where background pixels dominate most images. Two models were compared: one trained with uniform cross entropy loss and another trained with class weighted loss computed from training mask frequencies. Everything else remained constant. The goal was to test whether explicitly emphasizing rare classes improves segmentation performance. The results are summarized as below:

Class Weighting	mIoU	Pixel Accuracy	Class Label	Per-class Pixel Accuracy
Default	0.0431	0.7365	`background`	0.9866
			`bicycle`	0.0
			`car`	0.0
			`horse`	0.0
			`person`	0.1968
Re-weighted	0.0270	0.2053	`background`	0.2486
			`bicycle`	0.2974
			`car`	0.1338
			`horse`	0.1462
			`person`	0.0494

Introducing weights changed how the model distributed its predictions but did not improve overall performance (Figure 8). Pixel accuracy reduced from approximately 0.7 to 0.2. The weighting reduced background dominance but did not give the network enough signal to learn rare categories. To examine this more closely, I compared per class pixel accuracy for five categories in the table above: `background`, `bicycle`, `car`, `horse`, and `person`. The `background` accuracy fell sharply, `bicycle` showed small gains, and the other classes still remained low.

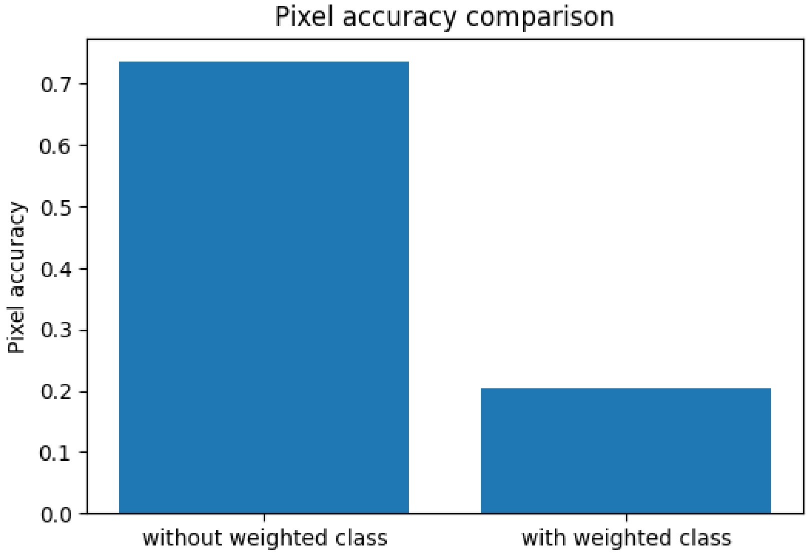

Figure 8: Effect of applying class weights on U-Net

4.3 Effect of Pretraining on DeepLabV3

This ablation examines how much ImageNet pretraining contributes to DeepLabV3's segmentation performance. The experiment kept everything else constant, but trained one model entirely from scratch while the main DeepLabV3 model used a pretrained ResNet-50 backbone.

Pretrained	mIoU	Dice Score	Pixel Accuracy	HD95
Yes	0.4545	0.6301	0.8794	6.0094
No	0.4186	0.5568	0.8578	8.4131

Training DeepLabV3 without pretraining still produced reasonable segmentations but decreased performance overall. Pixel accuracy, mIoU, and Dice were all slightly lower, and the model struggled relatively more with fine object boundaries. The effect of pretraining is more clearly visualized in the bar plot below. Pretraining provides strong low-level and mid-level visual features, allowing the full model to converge faster and handle more object categories. Without it, the network must learn these representations from scratch, which is difficult given the limited size of `VOC`.

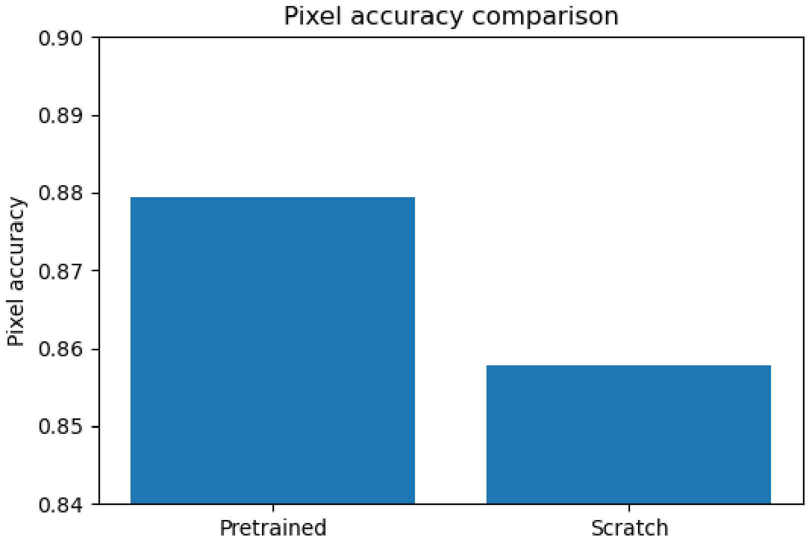

Figure 9: Effect of pretraining on DeepLabV3

5 Interpretations and Lesson Learned

Several patterns emerged across the models, visualizations, and quantitative metrics. DeepLabV3 performed the best consistently, which suggests that strong feature extraction and pretrained backbones matter more than architectural shape alone. Its higher mIoU of **0.4545** and pixel accuracy of **0.8794** show that multi-scale context and good initialization help the model generalize across diverse scenes. The bar plot of per-class accuracy shown in Figure 5 also illustrates that DeepLabV3 handled many everyday objects such as people, cars, and animals with decent confidence, which matches what we see in the prediction mosaics (Figure 4).

U-Net behaved differently. It segmented `background` well, with a per-class pixel accuracy of **0.9866**, but frequently missed smaller or infrequent classes. The predictions in its visual examples shown in Figure 2 usually show large area of `background` label but incomplete object boundaries. The ablation studies reinforce this pattern. Reducing the base channel size lowered performance even more, and the bar plot in Figure 7 comparing `64` vs `32` channels shows how limited capacity reduces pixel accuracy across most classes. Adding class weights shifted predictions toward rare categories but also decreased overall accuracy, suggesting that imbalance alone was not the only obstacle.

SAM displayed another behavior entirely. Its binary masks often grouped multiple objects together because it was not trained specifically for `VOC` semantic labels. The mosaics show smooth but broad foreground regions, and its pixel accuracy of **0.3559** reflects this. However, its HD95 of **0.0** indicates that when the foreground was correct, its boundary placement could be surprisingly precise. This shows that a general purpose model may still provide meaningful structural cues, even without fine tuning.

Across all experiments, several greater lessons became clear. Preprocessing and consistent evaluation strongly influence both qualitative and quantitative outcomes. Pretrained models deliver clear advantages when data is limited. Increasing model capacity helps only when relevant features exist in the data. Class reweighting cannot replace detailed supervision. Overall, these results and visualizations together helped clarify why some segmentation strategies succeed while others struggle, and how modeling choices directly shape final performance.

6 Conclusions

This project offered a hands-on perspective on semantic segmentation and highlighted how different modeling choices shape outcomes. DeepLabV3 set the performance baseline, U-Net provided a simple and interpretable reference point, and SAM illustrated what a general segmentation model can do without task specific training. The controlled ablations showed that capacity, initialization, and loss design all influence results, but not always in predictable ways. The complete implementation, including code, trained models, and generated figures, is available on GitHub.

The work also revealed clear areas for improvement. We cannot deny that U-Net might have performed better with longer training, richer augmentation, or more expressive feature extractors, especially given the dataset's diversity. SAM showed potential as well, and with additional guidance or adaptation it could align more closely with semantic labels. More data, alternative architectures, or different optimization strategies may also shift outcomes. Rather than simple fixes, these possibilities highlight how segmentation performance evolves as design choices, data, and computational budgets change.