Semantic Segmentation on PASCAL VOC 2012: A Comparative Study of U-Net and ResNet-50 U-Net

Mingwei Zhang, Junhong Tong
Department of Computer Science
Queens College, City University of New York
Flushing, NY 11367
mingwei.zhang67@qmail.cuny.edu junhong.tong16@qmail.cuny.edu

Abstract

Semantic segmentation performs classification at the pixel level, moving beyond the coarse bounding boxes of object detection. By assigning a label to every pixel, it generates semantic masks that capture fine-grained image structure. This work presents a comparative study of deep learning architectures for multi-class segmentation using the PASCAL VOC 2012 dataset. An 8-class subset is used to examine the effects of class imbalance and limited labeled data. A baseline U-Net trained from scratch is compared with an improved ResNet-50 U-Net that uses transfer learning. Both models use the same U-Net style decoder design and the same frequency-weighted cross-entropy loss to keep the training setup consistent, while the improved model replaces the baseline U-Net encoder with a ResNet-50 backbone pretrained on ImageNet. This design leverages deeper feature representations for object recognition. Experimental results show a clear performance difference: the ResNet-50 model achieves a mean Intersection over Union (mIoU) of 39.15%, representing a 125% relative improvement over the baseline value of 17.43%. The pretrained encoder also improves performance on more challenging classes, such as train, where the IoU increases by approximately a factor of five. These results highlight the importance of pretrained feature extractors for semantic segmentation when training data is limited.

1 Introduction

Semantic segmentation assigns a semantic label to every pixel in an image, enabling detailed scene understanding. Unlike image classification, which produces a single label for an entire image, semantic segmentation generates pixel-level predictions. This capability is essential for applications such as autonomous driving, medical image analysis, and robotic perception, where accurate object boundaries are required.

U-Net is a widely used segmentation architecture originally designed for biomedical image analysis. Its encoder-decoder structure with skip connections allows the network to combine high-level semantic features with low-level spatial information. However, training U-Net from scratch typically requires a large amount of labeled data. When the available dataset is limited, the model may struggle to learn strong feature representations.

To address this limitation, recent approaches have focused on transfer learning using pretrained encoders. The impact of replacing the U-Net encoder with a pretrained ResNet-50 backbone is examined for semantic segmentation on the PASCAL VOC 2012 dataset. The evaluation includes a baseline U-Net and a ResNet-50 U-Net, along with quantitative and qualitative comparisons of their performance.

2 Related Work

2.1 Semantic Segmentation

Fully Convolutional Networks (FCNs), introduced by Long et al. [1], were among the first architectures to perform end-to-end semantic segmentation. By removing fully connected layers and using only convolutional operations, FCNs enabled dense pixel-level predictions. This approach established a foundation for many subsequent segmentation models.

2.2 U-Net Architecture

U-Net was proposed by Ronneberger et al. [2] and features a symmetric encoder-decoder design with skip connections between corresponding layers. These skip connections help recover spatial details lost during downsampling. Although initially developed for biomedical imaging, U-Net has since been adapted for a wide range of general segmentation tasks.

2.3 Transfer Learning with Pretrained Encoders

ResNet, introduced by He et al. [3], uses residual connections to enable the training of very deep neural networks. Networks pretrained on large datasets such as ImageNet learn strong and general visual features. When applied to segmentation tasks, pretrained encoders often improve convergence speed and performance, particularly when training data is limited.

3 Methodology

3.1 Baseline Architecture: Standard U-Net

Our baseline model follows the symmetric encoder–decoder topology proposed by Ronneberger et al. [2]. The contracting path (encoder) consists of four stages, each comprising two 3×3 convolutional layers followed by Batch Normalization and ReLU activation. Spatial downsampling is performed via 2×2 max pooling with stride 2. The channel dimensions progressively double from 64 to 512. A bottleneck block further increases the channels to 1024 at the coarsest resolution (i.e., $1/16$ of the input resolution).

The expanding path (decoder) recovers spatial resolution using bilinear upsampling. Skip connections concatenate high-resolution encoder features with upsampled decoder features, and a subsequent convolutional block fuses them to refine object boundaries. Finally, a 1×1 convolution maps the 64-channel output to C class logits. The model is trained from scratch using PyTorch default initialization.

3.2 Improved Architecture: ResNet-50 Encoder

To leverage transfer learning, we propose an enhanced architecture that replaces the standard encoder with a ResNet-50 backbone [3] pre-trained on ImageNet. The encoder extracts hierarchical features at multiple scales: the stem output S_0 (64 channels, stride 2) and four residual stages S_1 to S_4 (with 256, 512, 1024, and 2048 channels at strides 4, 8, 16, and 32, respectively).

The decoder consists of four `UpBlocks` that progressively recover spatial resolution. Each `UpBlock` takes the feature map D_{in} from the previous decoder layer and a skip connection E_{skip} from the corresponding encoder stage:

$$D_{\text{out}} = \Psi\Big(\text{Concat}\big(\text{Upsample}(D_{\text{in}}),\ E_{\text{skip}}\big)\Big), \tag{1}$$

where $\Psi(\cdot)$ denotes a `ConvBlock` (two 3×3 convolutions with batch normalization and ReLU). A final bilinear upsampling layer restores the output to the original input resolution. This design effectively transfers discriminative features learned from ImageNet to the segmentation task, mitigating the limited training data of our PASCAL VOC subset.

3.3 Logarithmic Frequency-Weighted Loss

A critical challenge in this dataset is class imbalance, where background pixels vastly outnumber foreground objects. Optimizing a standard cross-entropy loss often leads to biased solutions that underrepresent rare classes (e.g., *train*, *bus*). To address this, we introduce a **Logarithmic Inverse-Frequency Weighted Cross-Entropy Loss**.

Let f_c denote the frequency of class c, estimated from a subset of training images. We compute a balancing weight w_c for each class as:

$$w_c = \frac{1}{\ln(1.02 + f_c)}. \tag{2}$$

The constant 1.02 serves as a smoothing term to ensure numerical stability. The weights are then normalized by their mean to maintain a consistent loss scale:

$$\tilde{w}_c = \frac{w_c}{\frac{1}{C}\sum_{j=0}^{C-1} w_j}. \tag{3}$$

The final training objective is:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=0}^{C-1} \tilde{w}_c \cdot 1[y_i = c] \cdot \log(p_{i,c}), \tag{4}$$

where N is the number of valid labeled pixels (across the batch), y_i is the ground-truth class label, and $p_{i,c}$ is the predicted probability for class c. This loss encourages the optimization to allocate more learning capacity to under-represented classes.

4 Experimentation

4.1 Dataset Description

The experiments are conducted on the PASCAL VOC 2012 segmentation dataset [4].

The official VOC download source was unreliable during the project period, so the dataset was obtained from a publicly available mirror of `VOCtrainval_11-May-2012`. Course guidance suggested switching to MS COCO only if VOC could not be accessed, and the mirror source enabled completing the project using VOC 2012.

A subset of eight classes is used: background, person, car, dog, cat, motorbike, train, and bus. Although the official dataset contains 1464 training images and 1449 validation images, the dataset is filtered to include only images containing at least one selected foreground class. This results in 873 training images and 850 validation images. All images are resized to 256 $\times$ 256 pixels. The dataset shows significant class imbalance, with background pixels accounting for approximately 60–80% of most images.

4.2 Data Augmentation

To improve generalization from the limited annotated samples in PASCAL VOC, we implement a robust augmentation pipeline $\mathcal{T}$. During training, both the input image and label map undergo synchronized geometric transformations: (1) **Random Resized Crop**, producing patches of 256×256 pixels with scale factors sampled from $\mathcal{U}(0.8, 1.0)$ and aspect ratios from $\mathcal{U}(0.9, 1.1)$; and (2) **Random Horizontal Flipping** with probability $p = 0.5$. Additionally, photometric distortions are applied solely to the input images, including brightness (± 0.15), contrast (± 0.15), and saturation (± 0.10) jittering, to simulate diverse lighting conditions. All inputs are normalized using ImageNet statistics ($\mu = [0.485, 0.456, 0.406]$, $\sigma = [0.229, 0.224, 0.225]$). During validation, images are resized to 256 $\times$ 256 without augmentation.

4.3 Setup

Our experiments are conducted on a single NVIDIA Tesla T4 GPU using Google Colab. We train both the baseline and improved architectures for 30 epochs using the AdamW optimizer (lr = 2×10^{-4}, weight decay = 1×10^{-4}) with a constant learning rate schedule. A batch size of 4 is employed, and Automatic Mixed Precision (AMP) is enabled to accelerate training. Input images are resized to 256×256 pixels. All experiments use a fixed seed of 42 to improve reproducibility.

4.4 Performance Metrics

To strictly align with the implementation logic defined in our `compute_metrics` function, we evaluate segmentation performance using two quantitative measures. For all metrics, pixels labeled with `IGNORE_INDEX` (255 in PASCAL VOC, representing *void*/ignore regions) are excluded from the calculation to ensure fair evaluation.

Pixel Accuracy (Global Accuracy). This metric computes the ratio of correctly classified pixels to the total number of valid pixels. It provides a general sense of the model's accuracy but can be misleading under severe class imbalance:

$$\text{PixelAcc} = \frac{\sum_i 1(\hat{y}_i = y_i) \cdot 1(y_i \neq \text{ignore})}{\sum_i 1(y_i \neq \text{ignore})}, \tag{5}$$

where $\hat{y}_i$ is the predicted label and y_i is the ground-truth label for pixel i, and ignore denotes `IGNORE_INDEX`.

Mean Intersection over Union (mIoU). mIoU is the standard metric for semantic segmentation. It computes the Intersection-over-Union (IoU) for each class c and then averages across classes. The IoU for class c is defined as:

$$\text{IoU}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c + \text{FN}_c}, \tag{6}$$

and the mean IoU is:

$$\text{mIoU} = \frac{1}{|C'|} \sum_{c \in C'} \text{IoU}_c, \tag{7}$$

where C' denotes the set of classes that yield a non-zero union (i.e., classes for which $\text{TP}_c + \text{FP}_c + \text{FN}_c > 0$). Classes with zero union are excluded to avoid undefined values. This metric penalizes both false positives and false negatives, providing a robust measure of segmentation quality across object categories.

4.5 Validation Process

Training is performed on the filtered VOC 2012 training split, and evaluation is performed on the filtered VOC 2012 validation split. After each epoch, the model is switched to evaluation mode and metrics are computed on the full validation loader (no gradient updates). Mean IoU is computed per mini-batch and then averaged across all validation batches to obtain the final validation mIoU reported for each epoch. The results reported in Table 1 correspond to the final trained checkpoint after 30 epochs.

4.6 Results

Overall and per-class validation performance for both models is reported, followed by training curves that illustrate convergence behavior.

Table 1: Overall Performance Comparison

Method	Pixel Acc	mIoU	Improvement
Baseline U-Net	0.5957	0.1743	N/A
ResNet-50 U-Net	0.8510	0.3915	+124.6%

Table 1. This table reports final validation performance after 30 training epochs, showing that the ResNet-50 U-Net improves pixel accuracy (0.5957 to 0.8510) and mIoU (0.1743 to 0.3915, +124.6%) compared to the baseline U-Net.

Table 2: Per-Class IoU Comparison on the Validation Set

Class	Baseline U-Net IoU	ResNet-50 U-Net IoU
background	0.616	0.845
person	0.246	0.587
car	0.063	0.224
dog	0.109	0.276
cat	0.094	0.345
motorbike	0.100	0.209
train	0.041	0.204
bus	0.105	0.266

Table 2. This table reports batch-averaged per-class IoU on the validation set, showing consistent improvements across all classes, with the largest gains for *person* (0.246 to 0.587) and *background* (0.616 to 0.845), and clear improvements for *car* and *train*.

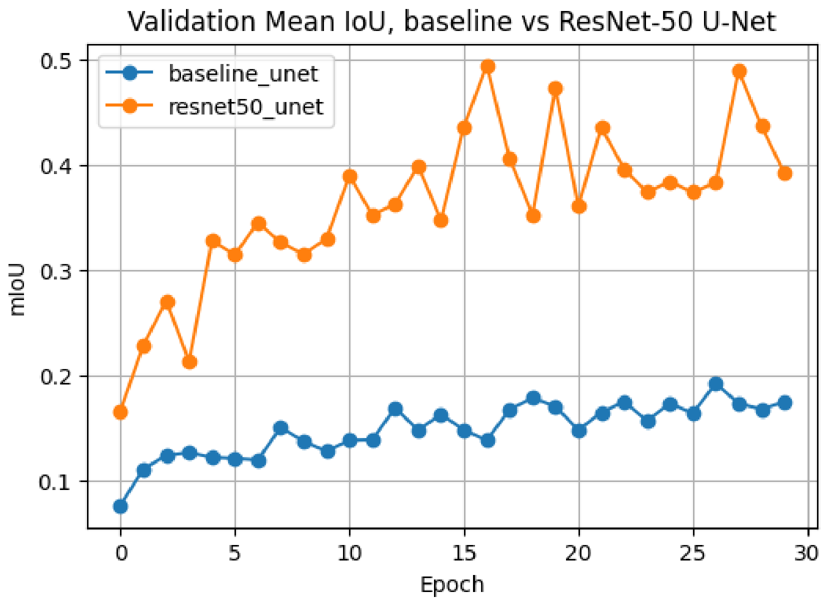

Figure 1: Validation mean IoU across epochs for the baseline U-Net and the ResNet-50 U-Net. The ResNet-50 U-Net reaches a higher mIoU early and remains consistently above the baseline, indicating stronger feature learning and better generalization throughout training.

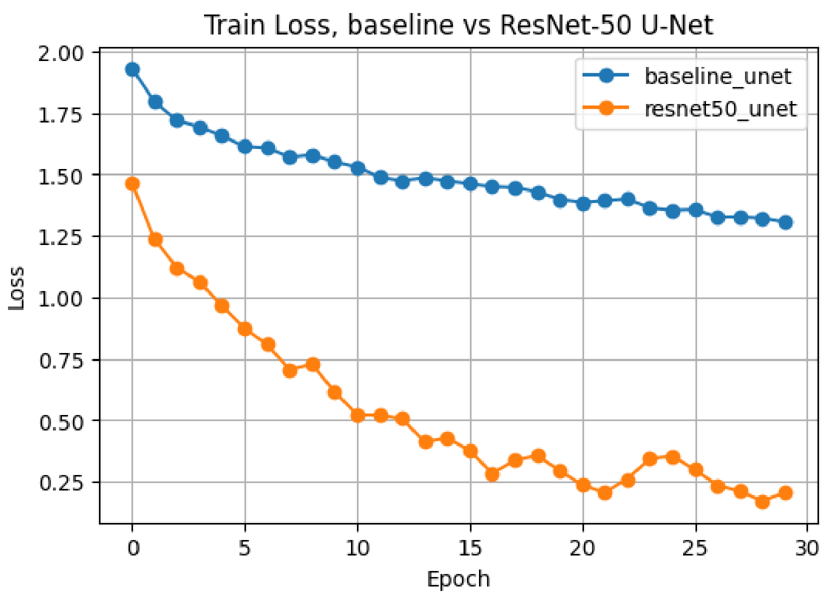

Figure 2: Training loss across epochs for both models. The ResNet-50 U-Net shows faster loss reduction and a substantially lower final loss, consistent with improved optimization from using a pretrained encoder rather than learning low-level features from scratch.

Overall, the pretrained ResNet-50 encoder leads to faster convergence and consistently stronger segmentation quality. The ResNet-50 U-Net maintains higher validation mIoU throughout training (Figure 1) and reaches a lower final training loss (Figure 2), which aligns with the large overall mIoU gain in Table 1 and the broad per-class IoU improvements in Table 2, especially on key foreground categories such as *person*, *dog*, and *cat*.

5 Discussion and Conclusion

5.1 Analysis of Results

The large performance gap between the two models is primarily due to the use of a pretrained encoder. The ResNet-50 backbone provides strong feature representations learned from large-scale image data, allowing the segmentation model to focus on task-specific learning rather than basic feature extraction. In contrast, the baseline U-Net must learn all features from scratch, which is challenging given the limited size and imbalance of the dataset.

In addition to quantitative metrics, Figures 3 and 4 provide qualitative comparisons. Each panel shows the input image, ground-truth mask, baseline U-Net prediction, and ResNet-50 U-Net prediction, using the same class-to-color mapping.

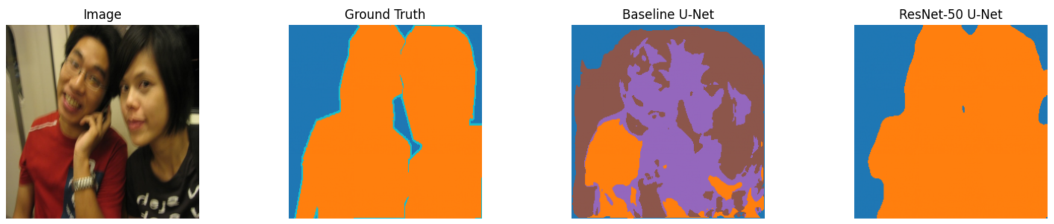

Figure 3: Qualitative comparison on a person-focused example. The baseline U-Net produces fragmented regions and assigns incorrect labels within the foreground, while the ResNet-50 U-Net recovers a cleaner foreground shape with clearer boundaries and fewer incorrect regions.

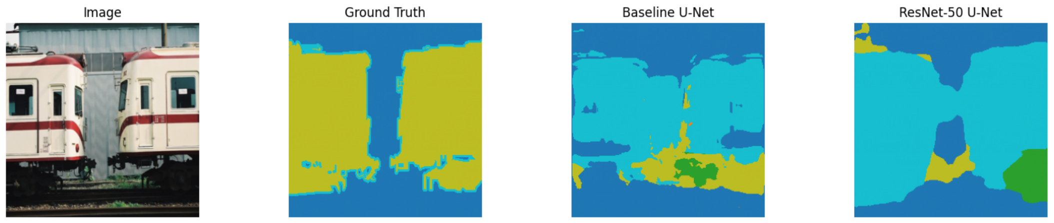

Figure 4: Qualitative comparison on a train example. The baseline U-Net shows confusion between object and background regions and introduces noisy patches, while the ResNet-50 U-Net yields larger contiguous object regions and improved separation between the train and surrounding background.

The low mIoU achieved by the baseline U-Net highlights the difficulty of training deep segmentation models without pretraining. These results reinforce the importance of transfer learning for semantic segmentation tasks with constrained data.

5.2 Limitations

Several limitations are present in the experimental setup. The input resolution of 256×256 restricts fine boundary accuracy, and higher resolutions could improve segmentation quality at the cost of increased computational requirements. In addition, only a subset of PASCAL VOC classes is evaluated, which limits the breadth of the analysis. Hyperparameter tuning is also constrained by available time and computational resources.

5.3 Future Work

Future improvements could include experimenting with more advanced architectures such as DeepLabv3+ [5], exploring deeper or alternative backbone networks, and incorporating techniques such as deep supervision or multi-scale inference. Extending the experiments to the full 21-class PASCAL VOC benchmark would provide a more comprehensive evaluation.

5.4 Conclusion

This experiment compares a vanilla U-Net and a ResNet-50 U-Net for semantic segmentation on the PASCAL VOC 2012 dataset. The ResNet-50 U-Net achieves a mean IoU of 0.3915, substantially outperforming the baseline U-Net, which attains an mIoU of 0.1743. These results indicate that incorporating pretrained encoders through transfer learning is an effective approach for improving segmentation performance when training data is limited.

Appendix: Team Member Contributions

Both authors jointly contributed to dataset preparation, model implementation, training and testing, evaluation metric design, and result visualization. Mingwei Zhang primarily led the implementation of the segmentation models, training pipeline, and experimental setup. Junhong Tong primarily contributed to the literature review, experimental analysis, and report writing. They collaborated closely on model design decisions, debugging, testing, and review of the final report.

References

[1] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *CVPR*.

[2] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*.

[3] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*.

[4] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. L. (2010). The PASCAL visual object classes (VOC) challenge. *IJCV*.

[5] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *TPAMI*.