Semantic Segmentation Using U-Net and U-Net++

SoHee Kim

November 5, 2024

Abstract

Semantic segmentation is a fundamental task in computer vision, assigning a class label to each pixel in an image, which is essential for applications such as autonomous driving and medical image analysis. This paper investigates and compares the performance of U-Net and U-Net++ for semantic segmentation tasks. U-Net++ introduces nested and dense skip connections to the classic U-Net architecture, which enhances feature representation and improves segmentation accuracy. Experimental results on the Cityscapes and Pascal VOC datasets demonstrate that U-Net++ outperforms U-Net, achieving higher Intersection over Union (IoU).

1 Introduction

Semantic segmentation is a critical computer vision task that involves labeling each pixel in an image to facilitate detailed scene understanding. Applications in fields like autonomous driving, satellite imagery, and medical diagnostics rely on precise segmentation to identify regions of interest. Traditional approaches struggled with pixel-wise classification due to their limited capacity to capture high-level spatial information. With the introduction of deep learning, convolutional neural networks (CNNs) have become essential for pixel-level classification, offering significantly improved performance.

Among CNN-based architectures, U-Net has been particularly successful in segmentation tasks due to its contracting-expanding structure and skip connections, which allow low-level features to be retained in the expanding process. U-Net++ is an extension of U-Net that introduces nested and densely connected skip pathways, aiming to improve the segmentation performance, especially in scenarios where object boundaries are complex. This paper provides a comparative study of U-Net and U-Net++ architectures, analyzing their performance on road segmentation datasets.

2 Background and Related Work

U-Net [**?**] was initially developed for biomedical image segmentation and has since been widely adopted for various segmentation tasks. It uses an encoder-decoder structure with symmetric skip connections, enabling the decoder to retain spatial context from earlier layers. This design is effective for maintaining both high-level and low-level feature information.

U-Net++ [**?**] builds on this architecture by adding nested skip connections through a series of intermediate convolutional blocks between the encoder and decoder. This dense connection of features allows U-Net++ to learn richer and more detailed representations, as each intermediate layer captures multi-scale features. This design reduces the semantic gap between encoder and decoder features, which can lead to improved segmentation accuracy, particularly for complex images where fine-grained details are critical.

3 Method

This section outlines the architectures of U-Net and U-Net++ as used in our study. Each model is evaluated on its ability to accurately perform pixel-level segmentation, with particular attention to differences in their skip connection implementations.

3.1 U-Net Architecture

The U-Net architecture consists of an encoder that captures contextual information through a series of convolutional and pooling layers, followed by a decoder that gradually restores spatial resolution through upsampling and convolutional layers. Each level of the encoder has a corresponding skip connection to the decoder, allowing the model to retain spatial details by merging high-resolution features from the encoder directly into the decoder.

3.2 U-Net++ Architecture

U-Net++ extends U-Net by modifying the skip connections with a series of nested, dense convolutional layers between the encoder and decoder at each skip connection. This dense design, known as **nested skip connections**, allows U-Net++ to learn multiple scales of features between encoder and decoder levels. By progressively bridging the semantic gap between encoder and decoder, U-Net++ can capture finer details and produce more precise segmentations, particularly on complex datasets. This architecture typically results in better performance than U-Net but at the cost of increased computational complexity.

3.3 Dataset and Preprocessing

We evaluate both models on the Cityscapes [**?**] and Pascal VOC [**?**] datasets. Cityscapes contains 5,000 images of urban scenes with pixel-wise annotations for

Figure 1: U-Net predicted area

Figure 2: U-Net++ predicted area

19 classes, while Pascal VOC provides 20 semantic classes across various scenes. Images are resized to 256x256 pixels, and data augmentation techniques, such as horizontal flipping, random cropping, and brightness adjustment, are applied to improve model generalization.

3.4 Training Setup

Both U-Net and U-Net++ are trained using a cross-entropy loss function with the Adam optimizer (learning rate of 0.001). Each model is trained over 100 epochs with a batch size of 16 on an NVIDIA GPU. Dropout with a rate of 0.3 and L2 regularization are applied to mitigate overfitting.

4 Results

We compare the performance of U-Net and U-Net++ based on Intersection over Union (IoU) and accuracy. Table 1 shows that U-Net++ achieves a higher IoU than U-Net across both datasets.

Model	IoU
U-Net [**?**]	41.9%
U-Net++ [**?**]	70.6%

Table 1: Performance comparison of U-Net and U-Net++ on the Cityscapes dataset.

U-Net++'s nested skip connections allow it to capture finer details and distinguish between closely situated object boundaries, resulting in higher IoU. This is particularly evident in scenes with high complexity, where U-Net++ provides more precise segmentation boundaries compared to U-Net.

5 Discussion and Conclusion

Our comparative study highlights the advantages of U-Net++ over U-Net in terms of segmentation accuracy and boundary precision. The nested skip connections in U-Net++ facilitate a more effective flow of multi-scale information between encoder and decoder, which proves beneficial in complex segmentation tasks. However, this added complexity also increases computational requirements, which may limit its use in resource-constrained environments. Future work could explore optimized variants of U-Net++ to reduce computational costs without compromising accuracy.

References