Semantic Segmentation of CamVid Dataset using U-Net

Uday Sankar Mukherjee 12021002016053

November 24, 2023

1 U-Net Architecture

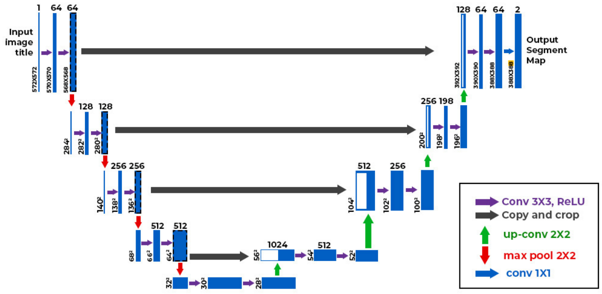

Figure 1: U-Net Architecture

The U-Net architecture consists of a contracting path, a bottleneck, and an expansive path. The contracting path captures context and spatial information through convolutional and pooling layers, while the expansive path enables precise localization using transposed convolutions. Skip connections between corresponding layers in the contracting and expansive paths help preserve spatial information.
The architecture is divided into the following key components:

1. Convolution Operation The convolution operation applied to an input tensor X with a filter W and bias b can be represented as:

$$Z = W * X + b$$

Here, Z is the output tensor.

2. Rectified Linear Unit (ReLU) Activation The ReLU activation function is commonly used to introduce non-linearity to the model:

$$A = \max(0, Z)$$

Where A is the activated output.

3. Max-Pooling Operation Max-pooling is typically employed to downsample the spatial dimensions of the tensor:

$$Y = \text{MaxPool}(X)$$

4. Transposed Convolution (Upsampling) Transposed convolution is used for upsampling:

$$Y = \text{ConvTranspose}(X)$$

5. Skip Connection A skip connection is represented as the concatenation of feature maps:

$$Y = \text{Concat}(X_{\text{skip}}, X_{\text{current}})$$

6. Softmax Activation The final layer often utilizes the softmax activation function for multi-class segmentation:

$$Y = \text{Softmax}(X)$$

7. Loss Function (Cross-Entropy) The cross-entropy loss for segmentation tasks is calculated as:

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C}\left[Y_{ij}\log(\hat{Y}_{ij})\right]$$

Where:
N is the number of pixels in the image,
C is the number of classes,

$$Y_{ij}$$

, is the ground truth label for pixel i and class j,

$$\hat{Y}_{ij}$$

, is the predicted probability for pixel i and class j.

8. Intersection over Union (IoU) IoU is a common evaluation metric for segmentation tasks, and it is calculated as:

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}$$

Where:
TP is the number of true positive pixels,
FP is the number of false positive pixels,
FN is the number of false negative pixels.
These equations provide a foundational understanding of the mathematical components of the U-Net architecture. Actual implementations may involve additional details such as batch normalization, dropout, and specific configurations for different layers.

1.1 Contracting Path

The contracting path comprises a series of convolutional blocks, each containing two convolutional layers with batch normalization and ReLU activation functions. Max-pooling is applied after each block to reduce spatial dimensions.

1.2 Bottleneck

The bottleneck consists of a double convolution block, which retains the most critical information in a compressed form.

1.3 Expansive Path

The expansive path consists of transposed convolutions and concatenation with the corresponding feature maps from the contracting path. Each block in the expansive path consists of two convolutional layers with batch normalization and ReLU activation.

1.4 Output Layer

The final layer employs a 1x1 convolution to produce the segmentation map with pixel-wise class predictions.

2 Loss Function and Optimization

The Focal Loss, a modification of the cross-entropy loss, is used as the loss function. Stochastic Gradient Descent (SGD) is employed as the optimization algorithm with momentum.

2.1 Results and Visualization

The training process is visualized by plotting the training and validation loss curves. Additionally, sample images, original masks, and predicted masks are displayed to assess the model's performance.

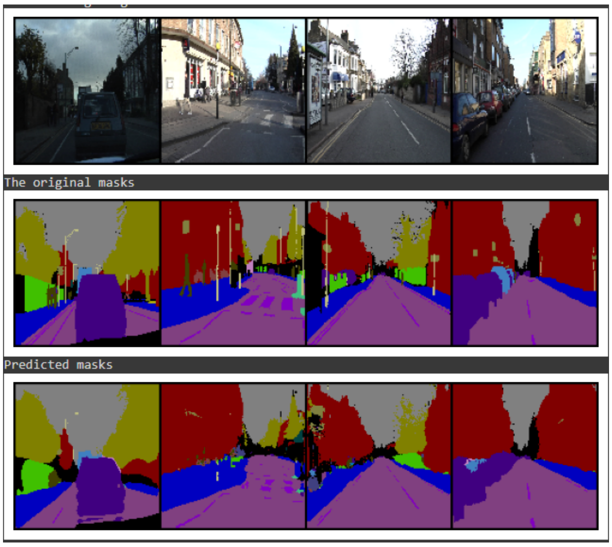

Figure 2: Segmentation Results

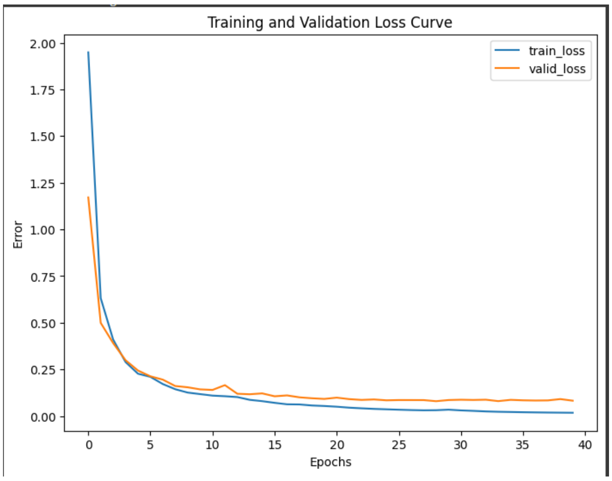

Figure 3: Loss Curve

3 Evaluation Metrics

The performance of a U-Net model is commonly evaluated using metrics such as Intersection over Union (IoU) and pixel accuracy. IoU measures the overlap between predicted and ground truth masks, providing insight into the model's ability to correctly identify object boundaries. Accuracy: 0.93

4 Conclusion

Semantic segmentation of the CamVid dataset using the U-Net architecture demonstrates the model's capability to accurately classify pixels into predefined classes. Experimentation with hyperparameters, additional data augmentation, or fine-tuning the model architecture may further improve performance. This documentation serves as a guide for implementing semantic segmentation tasks on similar datasets using the U-Net model.