# **Image Processing Project Report**

**Data Preparation**

* **Overview**

This report outlines the process of preparing data for a semantic segmentation task. The workflow involves loading images and masks, resizing them, normalizing the pixel values, applying data augmentations, and generating datasets for training and validation. The implementation is designed to handle custom color mappings for segmentation classes.

* **Dataset Details**
* 200 Training images.
* 70 Validation images.
* 150 Test images.

The dataset comprises training images, validation images, and corresponding segmentation masks stored in separate folders. Each image and mask pair is associated with matching filenames.

* **Image Preprocessing Workflow**

**Loading Images and Masks**

* Both are loaded with 3 channels to ensure consistency.

**Resizing**

* All images and masks are resized to a fixed size of ` (112, 112) `
* **Images**: Bilinear interpolation.
* **Masks**: Nearest-neighbor interpolation (to preserve class integrity).

**Normalization**

* **Images**: Pixel values are normalized to the range `[0, 1]` by dividing by 255.
* **Masks**: Masks are mapped to class indices using a custom color mapping.
* **Data Augmentation**

To enhance model generalization, data augmentations are applied during training:

**Augmentation Techniques**

**1. Horizontal Flipping**

**2. Vertical Flipping**

**3. Rotation**

**4. Brightness Adjustment**

* **Dataset Creation**

**Training Dataset**

* Augmented dataset is created using the `process\_and\_augment()` function, which chains preprocessing, normalization, and augmentation.
* Shuffling and batching are applied.

**Validation Dataset**

* The validation dataset uses the `process\_data()` function, which includes resizing and normalization without augmentation.
* Shuffling and batching are applied.

**Overview of Implemented Models and Modifications**

* **U-Net Model**

**Architecture**

The U-Net model is a well-known encoder-decoder-based architecture primarily designed for semantic segmentation. It incorporates symmetric skip connections between the encoder and decoder blocks, allowing detailed spatial information from the encoder to guide the decoder.

**Key Components**

**Double Convolution Block**

* Two consecutive convolution layers with kernel size 3x3, followed by batch normalization and ReLU activation.
* Optional dropout for regularization.

**Encoder Block**

* A double convolution block followed by a max-pooling layer for downsampling.

**Decoder Block**

* Upsampling using transposed convolutions, followed by concatenation with the corresponding encoder block and a double convolution block.

**Modifications**

* Dropout layers included as an optional regularization technique.
* L2 kernel regularization added to convolution layers.
* **Residual U-Net Model**

**Architecture**

The Residual U-Net builds upon the U-Net architecture by incorporating residual connections to improve gradient flow during training. This design aims to mitigate vanishing gradient issues and enhance feature learning.

**Key Components**

**Residual Convolution Block**

* Incorporates a skip connection that adds the input directly to the output of a series of convolutions.
* Two convolutional layers with batch normalization and ReLU activation.
* Adjusts the shortcut connection if input dimensions differ from output dimensions.

**Encoder Block**

* Residual convolution block followed by max pooling.

**Decoder Block**

* Transposed convolution for upsampling, concatenated with corresponding encoder features, followed by a residual convolution block.

**Modifications**

* Residual connections added to convolution blocks.
* Optional dropout layers for regularization.
* **SegNet Model**

**Architecture**

The SegNet model is another encoder-decoder-based segmentation model that differs from U-Net by not using skip connections. Instead, it relies on pooling indices during the encoder phase to guide the decoder for upsampling.

**Key Components**

**Encoder**

* Stacks of convolutional layers followed by batch normalization and ReLU activation.
* Max-pooling with stride 2 for downsampling.
* Retains pooling indices for use in the decoder.

**Decoder**

* Upsampling layers guided by pooling indices from the encoder.
* Convolutional layers followed by batch normalization and ReLU activation.

**Modifications**

* Dropout layers added for regularization.
* Batch normalization layers included after each convolution.

**Expected Benefits**

By using pooling indices instead of symmetric skip connections, SegNet simplifies the decoder design while maintaining spatial consistency. This may lead to reduced computational complexity compared to U-Net.

**Results and analysis from all experiments**

**Segnet Model**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Epochs** | **Learning Rate** | **Batch Size** | **Dropout** | **Optimizer** | **Validation Accuracy** |
| **60** | **0.0001** | **16** | **0.2** | **SGD** | **0.5337** |
| **60** | **0.0001** | **8** | **0.2** | **Adam** | **0.6302** |
| **60** | **0.00001** | **5** | **0.2** | **Adam** | **0.6186** |
| **50** | **0.0001** | **8** | **0** | **Adam** | **0.6327** |
| **40** | **0.0001** | **8** | **0** | **Adam** | **0.6309** |
| **30** | **0.01** | **5** | **0.4** | **SGD** | **0.4935** |
| **25** | **0.001** | **8** | **0** | **Adam** | **0.5542** |

**U-Net Model**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Epochs** | **Learning Rate** | **Batch Size** | **Dropout** | **Optimizer** | **Validation Accuracy** |
| **50** | **0.001** | **8** | **0** | **Adam** | **0.6980** |
| **50** | **0.0001** | **8** | **0** | **Adam** | **0.6600** |
| **35** | **0.1** | **24** | **0.3** | **Adam** | **0.4175** |
| **30** | **0.001** | **8** | **0.1** | **SGD** | **0.3141** |
| **20** | **0.1** | **12** | **0.3** | **Adam** | **0.4823** |
| **16** | **0.1** | **16** | **0.1** | **SGD** | **0.3334** |
| **15** | **0.1** | **8** | **0.2** | **Adam** | **0.4623** |

**Residual U-Net Model**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Epochs** | **Learning Rate** | **Batch Size** | **Dropout** | **Optimizer** | **Validation Accuracy** |
| **35** | **0.001** | **16** | **0** | **SGD** | **0.1688** |
| **35** | **0.0001** | **16** | **0.2** | **SGD** | **0.2274** |
| **20** | **0.001** | **8** | **0.2** | **SGD** | **0.3314** |
| **40** | **0.0001** | **8** | **0** | **Adam** | **0.7027** |
| **20** | **0.001** | **8** | **0.3** | **Adam** | **0.4823** |

**Conclusion**

This report concludes with an analysis of the implemented U-Net model for semantic segmentation and its variations. Each model, including U-Net, Residual U-Net, and SegNet, demonstrated distinct strengths and weaknesses:

1. **U-Net:**
   * Provided the best balance between accuracy and computational efficiency, achieving a validation accuracy of up to 0.6980 in experiments.
   * The symmetric skip connections ensured detailed spatial information was preserved during reconstruction.
   * The model's modularity facilitated easy tuning and experimentation.
2. **Residual U-Net:**
   * Enhanced gradient flow and feature learning with residual connections, improving performance for deeper architectures.
   * Achieved a peak validation accuracy of 0.7027 with optimized hyperparameters, slightly outperforming standard U-Net in specific configurations.
3. **SegNet:**
   * Reduced complexity by avoiding skip connections, relying on pooling indices for upsampling.
   * While computationally lighter, the model struggled with retaining fine-grained spatial details, leading to lower validation accuracy (e.g., 0.6302).

**General Insights:**

* Data augmentation techniques, including flipping, rotation, and brightness adjustments, played a critical role in enhancing model robustness.
* Regularization strategies such as L2 regularization and dropout effectively mitigated overfitting, particularly on smaller datasets.

These results underscore the importance of architecture selection and hyperparameter tuning in achieving optimal segmentation performance. The implemented U-Net model is a robust architecture for semantic segmentation, incorporating advanced regularization and normalization techniques. It is well-suited for the task with the provided dataset and can be easily adapted for other segmentation tasks with minor modifications.