Semantic Segmentation of Street Scenes: A Detailed Report

**1. Introduction**

This project involves the semantic segmentation of street scenes using deep learning. The primary goal is to label each pixel in an image with one of 8 predefined semantic classes: Background Clutter, Building, Road, Tree, Low Vegetation, Moving Car, Static Car, and Human.

**2. Data Preparation**

2.1 Data Resizing

The images in the dataset were resized to a consistent size of 256 × 256 pixels to ensure uniformity across the dataset and compatibility with the input dimensions required by the deep learning models. Both input images and their corresponding masks were resized using bilinear interpolation for images and nearest-neighbor interpolation for masks to preserve categorical labels.

2.2 Normalization

Before feeding the images into the models, normalization was applied. Each pixel value was scaled to the range [0, 1] by dividing by 255. This step helped accelerate model training by ensuring numerical stability and convergence during optimization.

2.3 Data Augmentation

To improve model generalization and reduce overfitting, data augmentation techniques were applied to the training set. The following transformations were used:

* Random horizontal flips
* Random rotations
* Random zooming
* Shear
* Shift

These augmentations were applied only to the input images while ensuring the corresponding masks remained consistent.

**3. Overview of Implemented Models**

**3.1 U-Net Architecture**

* Encoder-Decoder Structure: The U-Net consists of a contracting path (encoder) that captures context and an expansive path (decoder) that enables precise localization.
* Skip Connections: Skip connections link the encoder and decoder blocks, allowing fine-grained features from the encoder to guide the decoder.
* Loss Function: For this task, a weighted sparse categorical cross-entropy loss was used to handle the class imbalance across the 8 semantic classes.

**3.2 ResNet U-Net Architecture**

* Encoder (ResNet Backbone): The encoder uses a ResNet model.
* Decoder: Similar to the U-Net decoder, this part reconstructs the segmentation map from the encoded features while incorporating skip connections.
* Residual Connections: Residual blocks in the encoder help alleviate the vanishing gradient problem and enhance feature extraction.
* Loss Function: Sparse categorical cross-entropy was used to improve pixel-wise class prediction accuracy.

**4. Results and Analysis**

The performance of the models was evaluated using the following metrics:

* Intersection over Union (IoU) {standard: 51.9 %} {residual: 49%}
* Accuracy {standard:82.5 %} {residual: 81.8% }

* Precision {standard:67.3% } {residual: 64.3% }

* Recall {standard:61.7%} {residual: 60.2% }

**5. Conclusions and Insights**

* U-Net is a strong baseline for semantic segmentation, especially for datasets with limited annotations.
* Data augmentation and careful loss design are critical for achieving high accuracy on imbalanced datasets.

**6. Code Repository**

The code for this project is hosted in a public GitHub repository. The repository includes:

* Model Implementations: Python notebook for U-Net and ResNet U-Net.
* Repository Link: <https://github.com/Abdelrahman-Shamikh/image_processing_25>