Preprocessing Report for Image Segmentation on Cityscapes Dataset (**Subhasis Panda**)

1 Introduction

This report describes the preprocessing pipeline developed for the Cityscapes dataset as part of an image segmentation project. The objective was to simplify the label space, standardize image dimensions, and prepare the dataset for training and validation using a deep learning-based segmentation model.

2 Key Decisions

2.1 Class Simplification Strategy

The original Cityscapes dataset provides 30+ classes. Many of these are overly fine-grained or semantically redundant for standard segmentation tasks. Therefore, we adopted a class simplification strategy:

• Processed 3500 images across train/val splits.

• Defined a **valid classes** list of 6 useful categories, and background.(Classes are **background**, **road**, **building**, **vehicle**, **person**, **vegetation**, **sky**)

• Constructed a **label mapping** dictionary that maps original pixel-wise label values to a simplified set of continuous class indices.

2.2 Technical Choices

Key technical decisions made in the preprocessing pipeline:
• **Libraries Used**: PIL, NumPy, os, random, and argparse.
• **Directory Structure**:Preprocessed images and masks are saved in **Image Processed** and **Masks Processed**.
• **File Handling**: Consistent pairing of RGB images and their corresponding masks using matching filenames.
• **Automation**: The script can be run from CLI with customizable input/output paths and image resolution.

2.3 Normalizing and Resizing Strategy

• **Target Resolution**: All images and masks resized to (256, 256) (height × width) to ensure uniformity across the dataset and enable batch processing during training.
• **Image Normalization**:Images are normalized to a [0, 1] range by dividing pixel values by 255. This helps in stabilizing the training process and improving convergence speed.
• **Interpolation**: **RGB images**: bilinear interpolation to preserve visual quality. **Segmentation masks**: nearest-neighbor interpolation to retain integer class indices.
• **Output Format**:Normalized images are saved as .npy files to retain floating-point precision.Processed masks are saved as .png files for compatibility and visual inspection.

3 Edge Case Handling

The following strategies were considered for edge cases: • Missing Annotation Files: Check if GT file exists before processing, Log warning and skip image.
• Unknown Class IDs: Map unmapped classes → background (class 0), and Log occurrence.
• Image Processing Errors: Try-catch around image loading/mask generation, Log errors and continue.
• Directory Validation: Verify input directories exist before processing.
• Stats Tracking: JSON file records all edge cases per split. Includes counts for Missing annotations, Unknown classes, Failed processing, Total processed images.
• Overlapping Masks : For the Cityscapes dataset, overlapping masks are not an issue because of its annotation methodology: Pixel-Wise Exclusivity (each pixel has exactly one semantic label, annotations are mutually exclusive by design), so no overlapping class assignments in ground truth.

4 Processing Statistics

4.1 Example Outputs(Visualization)

Here are a few examples of the final outputs(original image(resized) + processed mask + overlays):

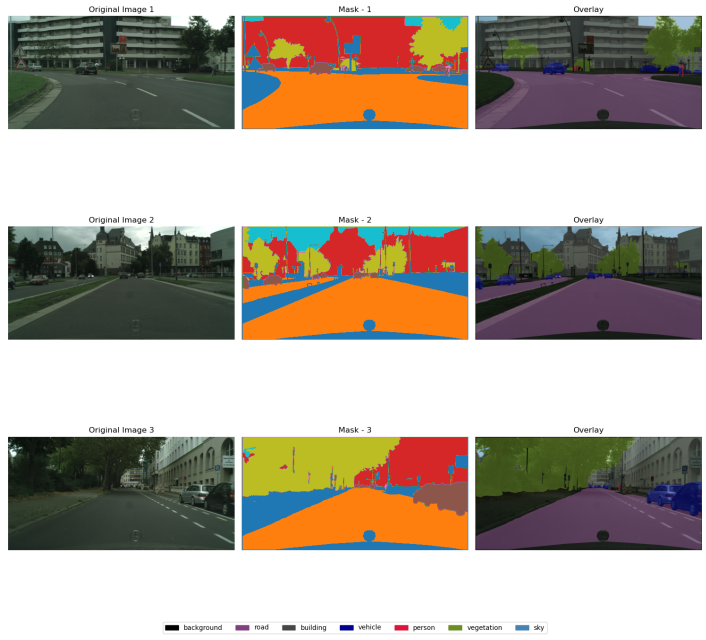
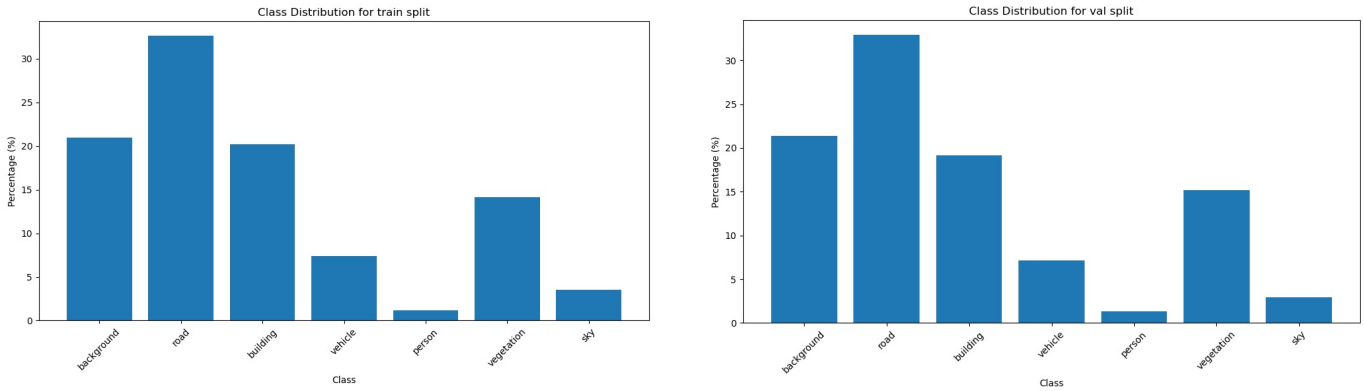


Figure 1: Sample Training Data, Corrosponding Mask, and Overlay Viuslization

4.2 Class Distribution



5 Reproducibility

GitHub Repo Link:https://github.com/subhasisp1/Semantic-Segmentation-Cityscapes
This GitHub Repo contains all the necessary Jupyter notebooks, Python scripts, stats, and processed images and masks that will be used for training/fine tuning our model.
Environment
• OS: Linux

- Python: Managed using **uv**
- Installation Steps:

```
# Clone repository
git clone <your-repo-link>
cd <your-repo-dir>

# Setup virtual environment using uv
uv venv
source .venv/bin/activate

# Install dependencies
uv pip install -r requirements.txt

# Run preprocessing script
python preprocess.py --input_dir ./leftImg8bit --label_dir ./gtFine --output_dir ./processed_data
```

- Dependencies :Listed dependencies in requirements.txt. (Pillow, numpy, argparse)

6 References

- Arulananth TS, Kuppusamy PG, Ayyasamy RK, Alhashmi SM, Mahalakshmi M, Vasanth K, et al. (2024) Semantic segmentation of urban environments: Leveraging U-Net deep learning model for cityscape image analysis. PLoS ONE 19(4): e0300767.https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0300767
- Claude, DeepSeek