Automatic Pavement Crack Segmentation around Manholes using U-Net 🤗

Hyunah Ko (2022144024)

1 Introduction

This project aims to build an AI pipeline for the automated semantic segmentation of pavement cracks around manholes in urban road environments. Manhole surroundings are structurally vulnerable points prone to complex failures such as radial cracking, which is a critical indicator of potential sinkholes. While traditional manual inspection is labor-intensive and subjective, naïve computer vision methods often fail due to road textures and shadows. In this project, I implemented a deep learning-based pipeline using U-Net and compared it with a heuristic baseline (Canny Edge Detection) to demonstrate the effectiveness of AI in handling these environmental challenges.

2 Task Definition

- **Task description:** The goal of this project is Semantic Segmentation, specifically detecting pavement cracks around manholes. The model takes a road surface image as input and classifies each pixel as either 'Crack' (1) or 'Background' (0).
- **Motivation:** Cracks around manholes are critical early warning signs of structural failure, often leading to sinkholes or potholes. However, detecting them is challenging because:
  1. **Visual Complexity:** Cracks are often thin and irregular, making them hard to distinguish from the rough texture of asphalt.
  2. **Environmental Noise:** Shadows cast by trees, buildings, or the manhole rim itself create strong edges that confuse traditional computer vision algorithms.
  3. **Inefficiency:** Manual inspection is labor-intensive, subjective. An automated AI pipeline is needed to prioritize maintenance efficiently.
- **Input / Output:**
  - **Input:** 512×512 grayscale images of manhole surroundings, cropped from high-resolution road surface images captured by survey vehicles.
  - **Output:** A binary segmentation mask of the same resolution (512×512), where pixels corresponding to cracks are labeled as 1 and the background as 0.
- **Success criteria:** The primary criterion for success is achieving a significantly higher IoU (Intersection over Union) score compared to a naïve baseline (Canny Edge Detector). Specifically, a "good" system should:
  - Effectively ignore high-frequency noise from asphalt texture.
  - Robustly distinguish between actual cracks and shadow boundaries.
  - Maintain the connectivity of thin crack lines in the segmentation mask.

3 Methods

This section includes both the naïve baseline and the improved AI pipeline.

3.1 Naïve Baseline

I implemented a Canny Edge Detector as the naïve baseline.

- **Method description:** The Canny algorithm detects edges by looking for local maxima of the gradient of the image. I applied Gaussian blurring to reduce noise and used hysteresis thresholding to identify crack edges.
- **Why naïve:** It is a heuristic method that relies solely on pixel intensity gradients without any learning process. It lacks the ability to understand the semantic context, meaning it cannot distinguish between structural cracks and other "edges" like road markings, shadows, or asphalt aggregates.
- **Likely failure modes:**
  - **Texture Noise:** It tends to detect the rough texture of asphalt as cracks.
  - **Shadow Boundaries:** It falsely identifies the sharp boundaries of shadows as structural cracks.

3.2 AI Pipeline

I designed an improved pipeline using a pre-trained U-Net model.

- **Models used:** U-Net with a ResNet34 encoder pre-trained on ImageNet. ResNet34 was chosen as it balances feature extraction and computational efficiency for the limited dataset size.
- **Pipeline stages:**
  1. **Preprocessing:** Images are resized to 512×512 and normalized using ImageNet statistics. CLAHE (Contrast Limited Adaptive Histogram Equalization) is applied to handle varying lighting conditions.
  2. **Training:** The model is trained using a Composite Focal-Dice Loss ($0.3 \times Focal + 0.7 \times Dice$) to address the severe class imbalance, as crack pixels occupy a very small portion of the image.
  3. **Optimization:** The Adam optimizer is used with a learning rate of $5e^{-5}$ and a ReduceLROnPlateau scheduler.
- **Design choices and justification:** U-Net was selected for its proven ability to segment fine, thin structures effectively. The combination of Focal and Dice loss ensures the model focuses on hard examples (cracks) rather than the dominant background.

4 Experiments

4.1 Datasets

- **Dataset source**: The dataset is derived from the "High-Resolution Road Surface Image Data" (G_A_V subset) provided by AI Hub[1].
- **Size & Preprocessing**: I filtered and cropped images containing manholes and manually annotated them using Roboflow. The base dataset consists of 206 images. To prevent overfitting, geometric (Rotation $\pm 15°$) and photometric (Brightness $\pm 20\%$) augmentations were applied.

- **Splits**: The final augmented dataset consists of 485 images total, split into 435 Train, 50 Validation, and 11 Test images.

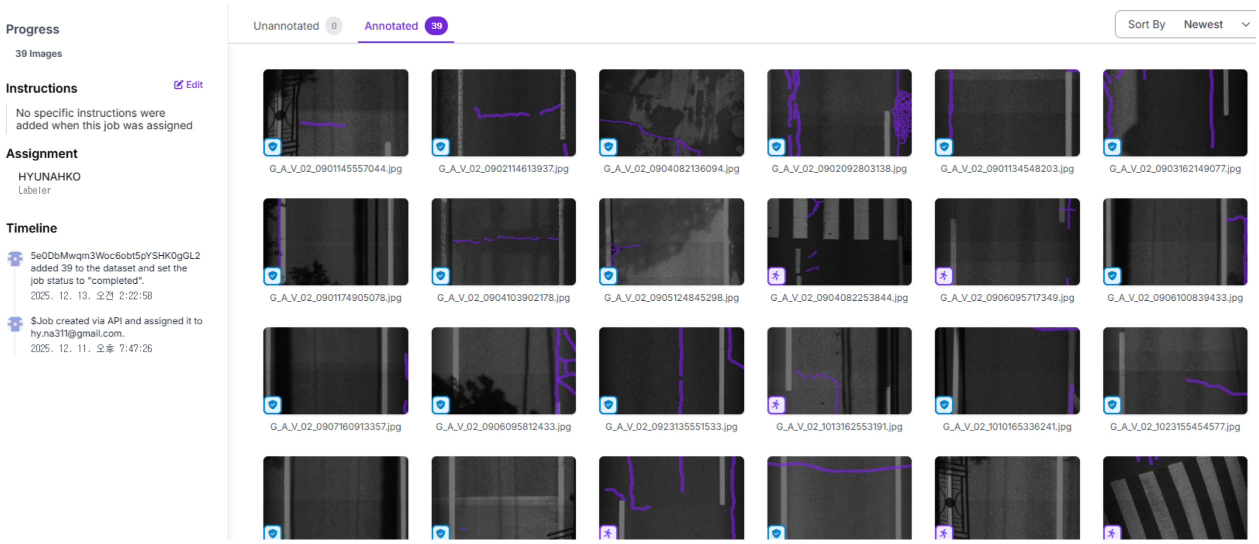

Figure 1: Data preprocessing and augmentation pipeline using Roboflow. The dataset was manually annotated for semantic segmentation.

4.2 Metrics

- **IoU (Intersection over Union):** The primary metric measuring the overlap between the predicted and ground truth areas.
- **Dice Coefficient (F1-Score):** Used to complement IoU, providing a balanced view of precision and recall for imbalanced data.

4.3 Results

The quantitative comparison between the baseline and the AI pipeline is summarized below:

Method	Average IoU	Average Dice
Baseline (Canny)	0.0246	0.0461
AI Pipeline	**0.2195**	**0.3145**

- **Analysis:** The AI pipeline achieved an IoU of 0.2195, which is approximately 9 times higher than the baseline (0.0246).

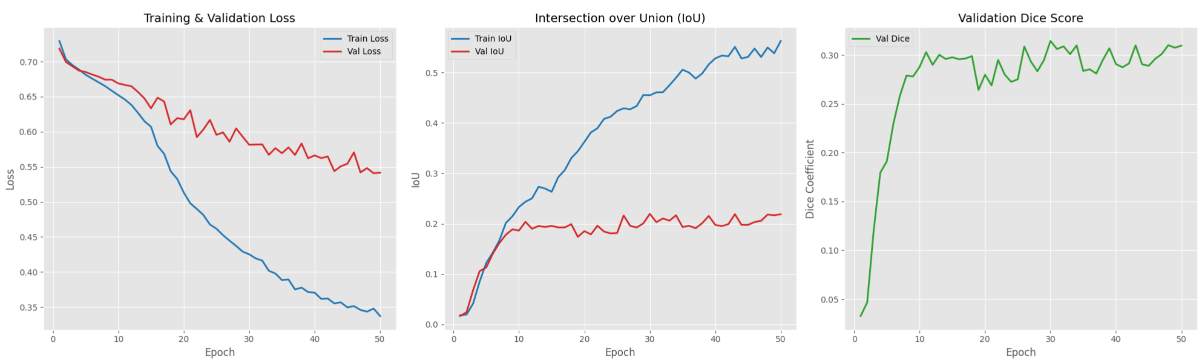

Figure 2: U-Net training and validation performance graphs over 50 epochs. The graphs show the trends of **(Left)** Loss, **(Middle)** IoU, and **(Right)** Dice Score. The gap between training and validation lines remains small, indicating stable learning without overfitting.

- **Qualitative Examples:**

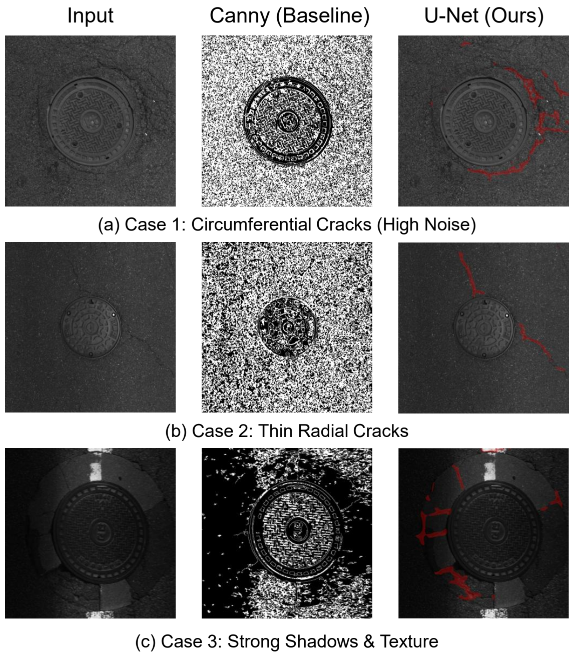

Figure 3: Qualitative comparison results under different environmental conditions. **(a)** Circumferential cracking where the baseline fails due to texture noise. **(b)** Thin radial cracking successfully segmented by the proposed U-Net. **(c)** Challenging shadow scenario where the U-Net demonstrates superior robustness compared to the baseline.

5 Reflection and Limitations

- **What worked better:** The AI pipeline (U-Net) significantly outperformed the heuristic approach. The use of Focal-Dice loss was particularly effective in forcing the model to learn thin crack features despite them being a minority class. The model successfully learned to ignore asphalt texture, which was the main failure point of the baseline.
- **What failed / Limitations:** Despite the relative improvement, the absolute IoU (0.22) is still low compared to general semantic segmentation tasks. This is due to the extreme thinness of cracks; a deviation of just 1-2 pixels significantly drops the IoU score. Additionally, the baseline (Canny) proved to be practically unusable for this task due to its sensitivity to shadows.
- **Future Work:** I plan to implement DeepLabV3+ with Boundary-weighted Loss to further refine edge precision. Furthermore, I aim to develop a morphological analysis algorithm to classify crack types (radial vs. circumferential) for a more detailed structural risk assessment.

References

[1] AI Hub. High-resolution road surface image data. https://aihub.or.kr/aihubdata/data/view.do?dataSetSn=71781, 2024. Accessed: 2024.