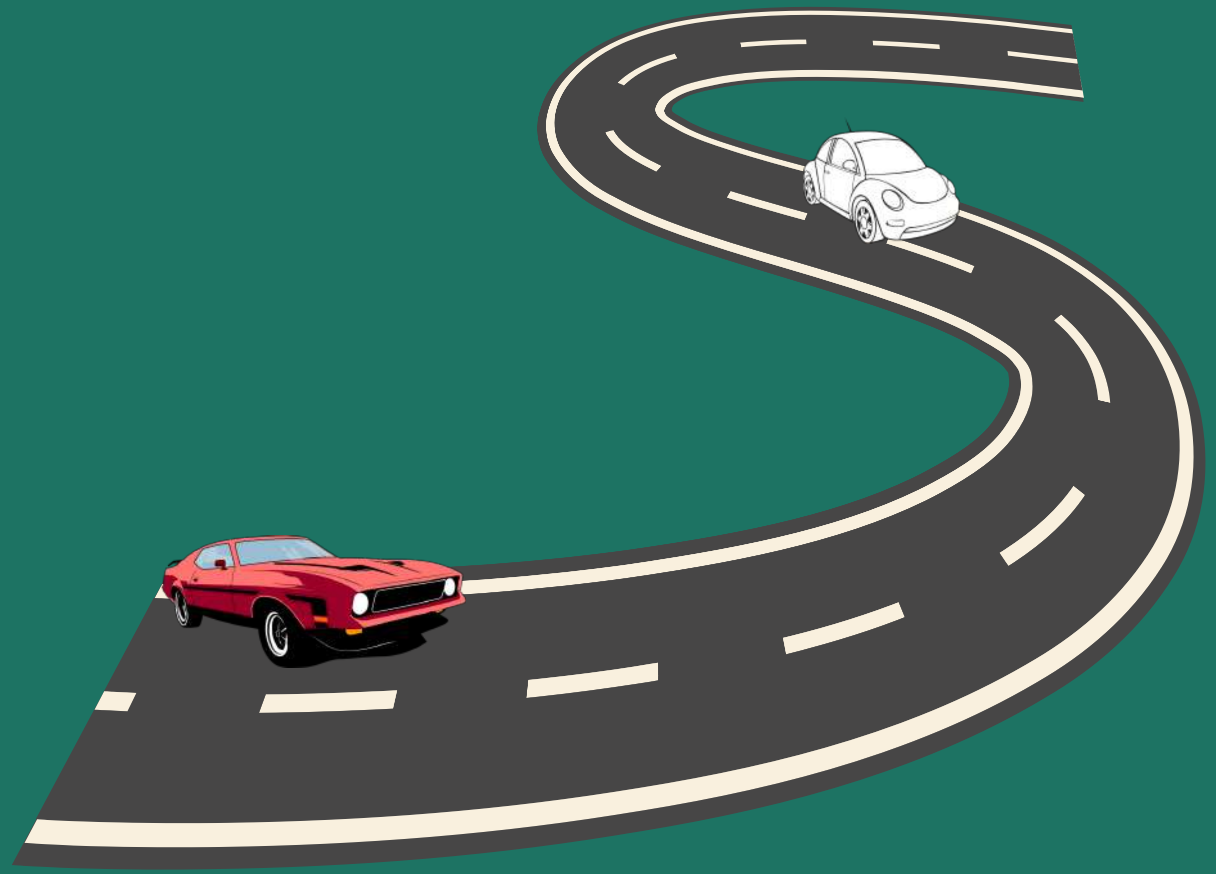
BINUS UNIVERSITY

In Partnership with ASO COLLEGE GROUP JAPAN

BINUS ASO SCHOOL OF ENGINEERING

Road Semantic Segmentation

Group 1

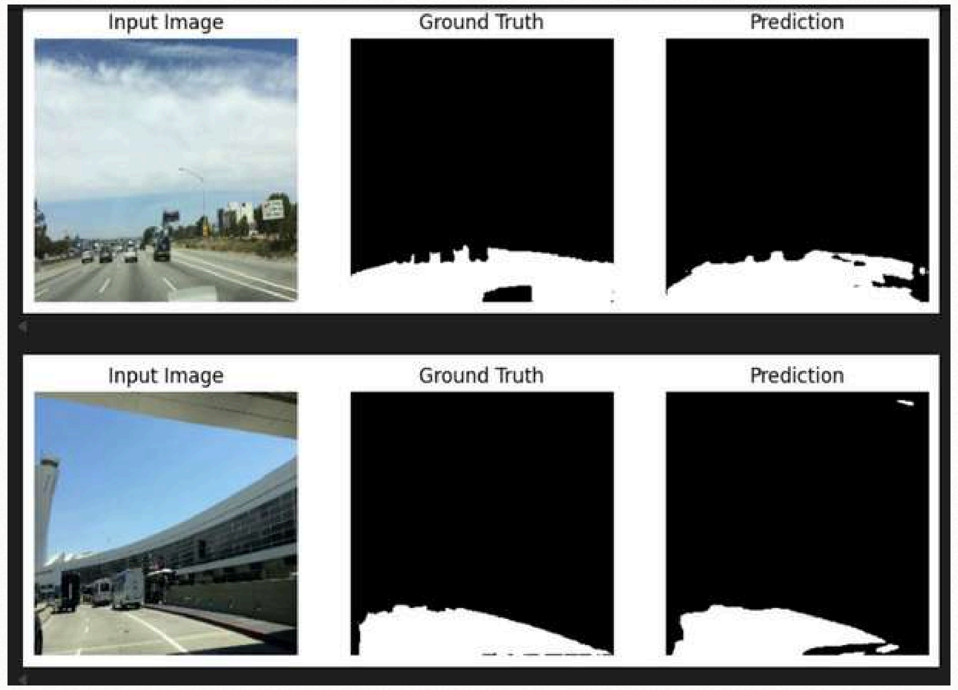


Computer Vision for Vehicle

TEAM

- Andhika Vidyatara Dwinatha Irianto **2702314416**
- Joaquin Tristan Susanto **2702230165**
- Rafael Skolast Tedjosurya **2702332085**
- James Austin Widjaya **2702380342**

INTRODUCTION



Using semantic segmentation for road segmentation with the use of U-Net model to accurately do pixel-level prediction for each image. Additionally using IoU (Intersection of Union) evaluation with a value >0.5 overlap between predicted and ground truth segmentation masks.

OBJECTIVE

Our project aims to detect roads using semantic segmentation on a live camera input.

The model should be capable of detecting roads on varying lighting environments, as we plan to identify drivable roads under daytime and nighttime.

OBJECTIVE

We start by building a U-Net deep learning model to detect road areas over hundreds of images, and use multiple datasets from different sources as input where each of them have matching masks (grayscale img).

DATASET

- **BDD100K**

  A road segmentation dataset sourced from Berkley DeepDrive.

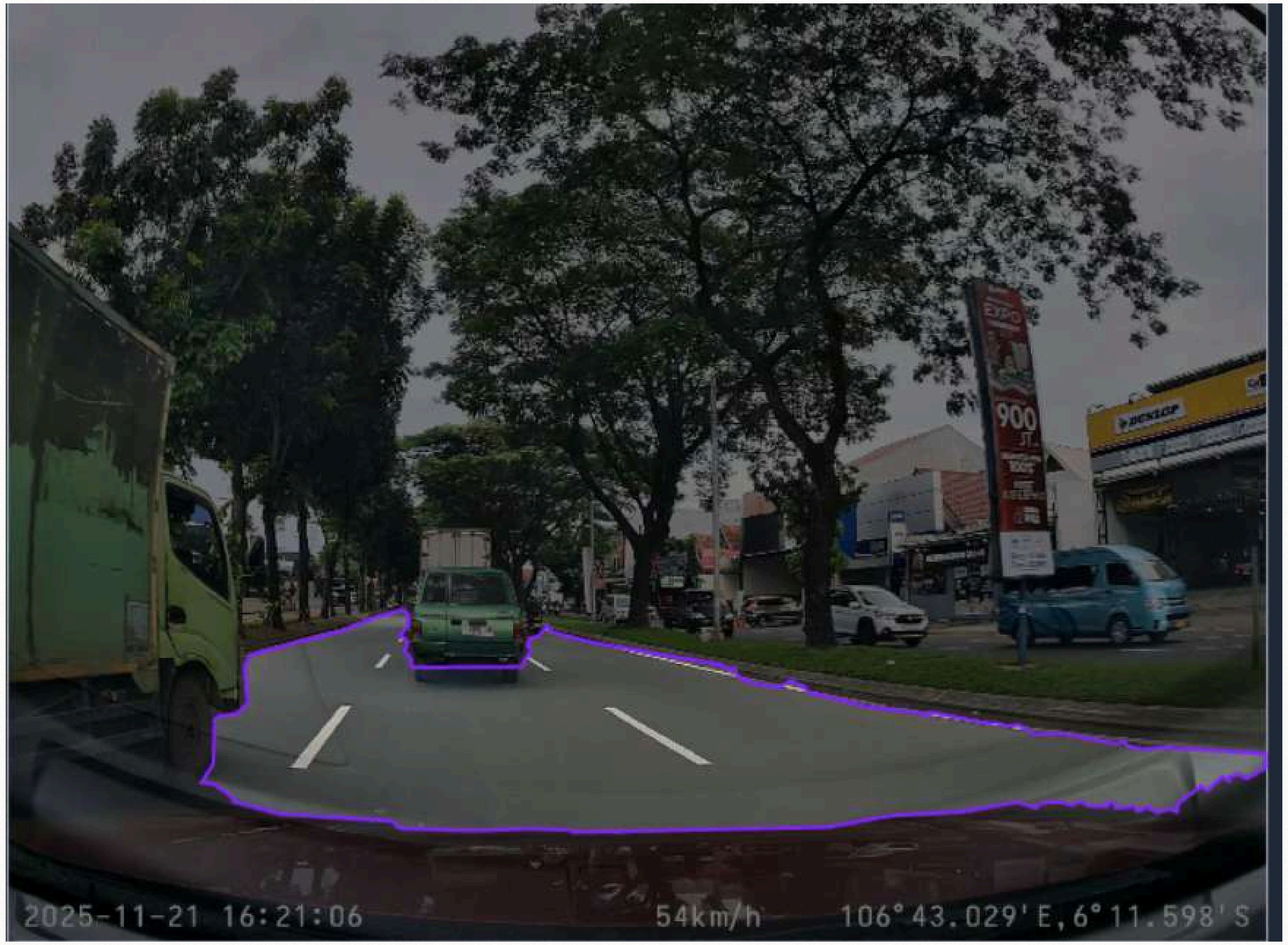
- **Dashcam**

  A dataset consisting of our dashcam footage strolling around Alam Sutera.

BDD100K

Dashcam

IMPLEMENTATION PLAN



Manually matching masks in Roboflow for each and every dataset image

MASKING

Masks for Supervised Deep Learning

DESIGN PIPELINE

1
Extract Dataset
Unzips and sets *image_base* / *label_base* for images and masks

2
Loading+Pre-processing
Reads image and masks, resizes to 256x256, scales images and masks to [0,1].

3
Augmentation
Apply horizontal flip and random brightness.

4
Dataset
Builds dataset from sorted file lists, map loaders, shuffles, batches, and prefetches.

5
Model
U-Net style encoder/decoder (3 encoder/decoder blocks, small channel counts 16 → 32 → 64) with final sigmoid output for binary segmentation.

6
Compile & Metric
Uses adam, accuracy, and IoU metric computer on predictions > 0.5

7
Train & Save
Fits training and validation for EPOCHS and saves model to .keras file

8
Visualization
Show sample inputs, ground truth masks, and thresholded predictions

1
Feature Extraction
To learn feature maps: Vertical/Horizontal edges, Texture patterns, Road curvatures, etc.

→

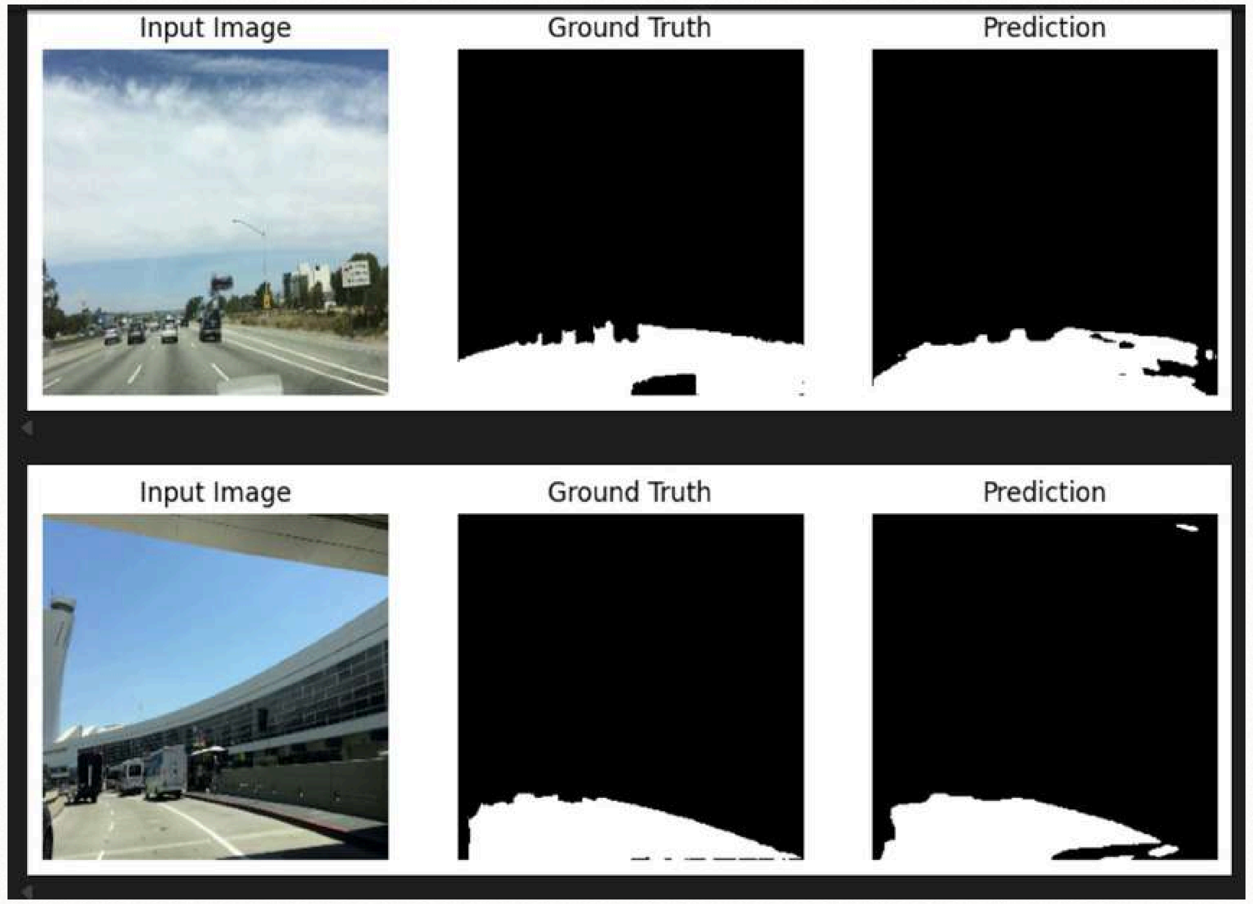
2
ReLu Activation
Makes features non-linear to detect road boundaries and lane lines

→

3
MaxPooling
Reduces spatial size where it keeps important features, while removing unnecessary details.

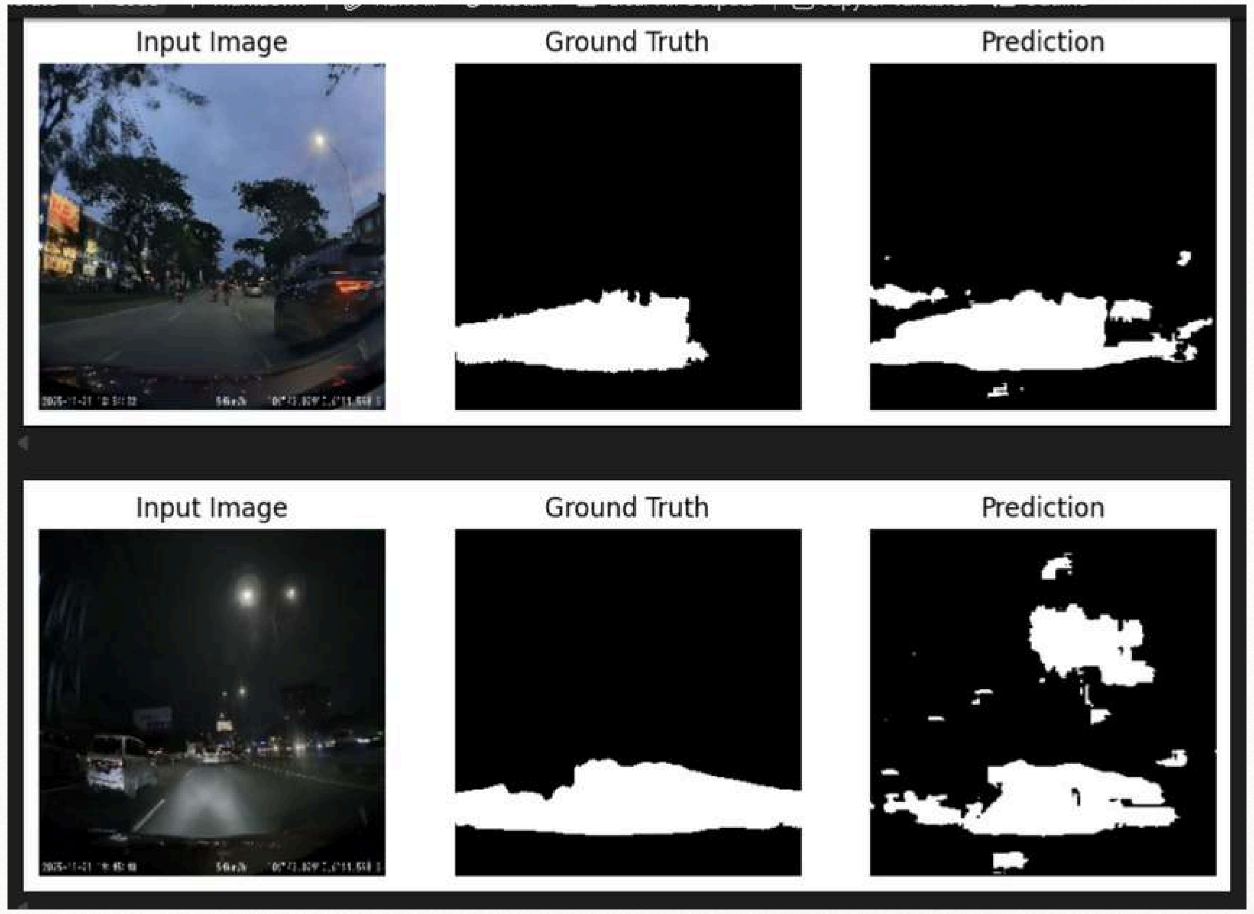
PRELIMINARY DESIGNS

V1 (ORIGINAL)



BDD100K 100IMG Dataset

- Accuracy = 94.86%
- IoU Metric = 67.92%
- Val Accuracy = 95.17%



*BDD100K+Dashcam 7*00IMG Dataset

- Accuracy = 96.82%
- IoU Metric = 73.96%
- Val Accuracy = 94.09%

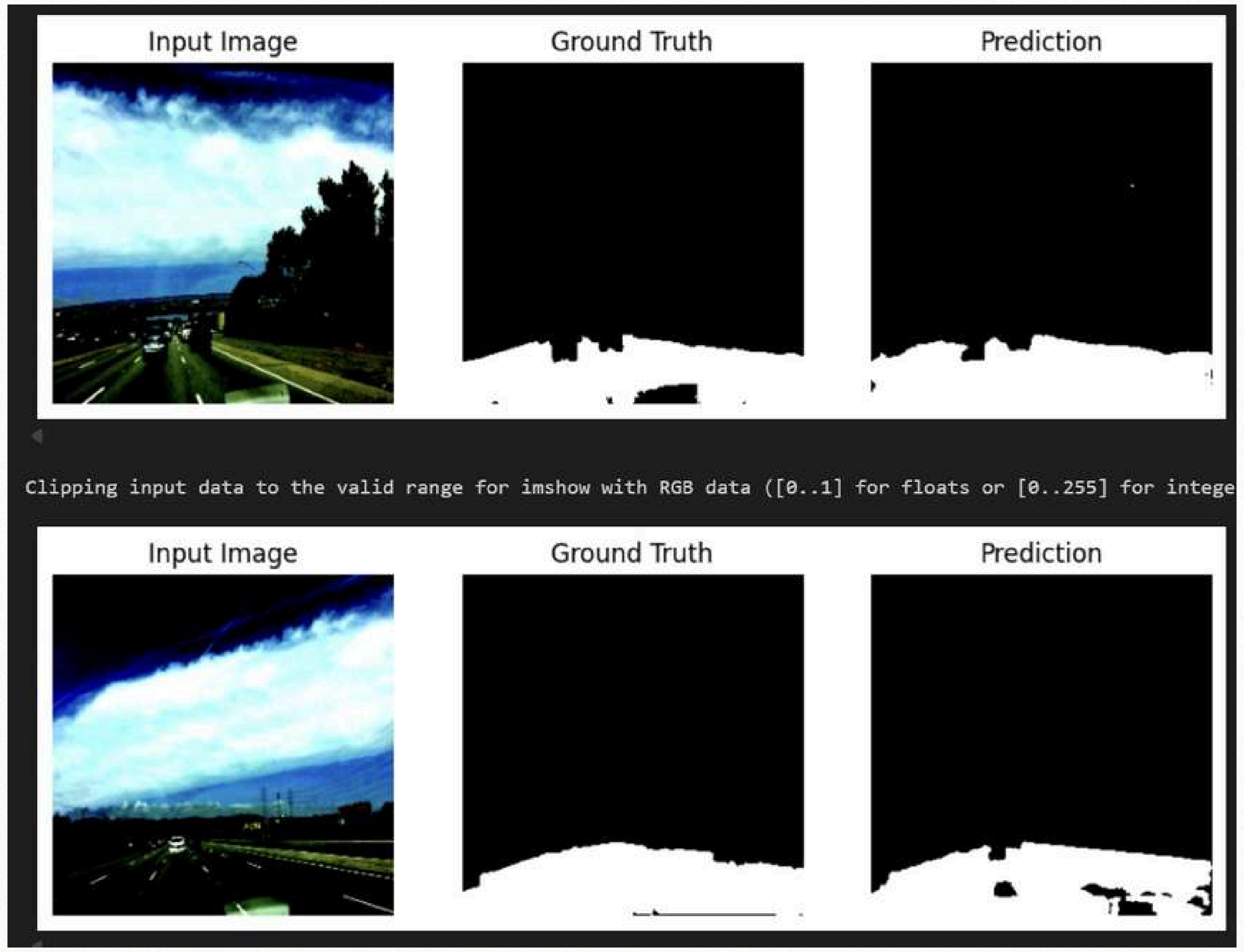
Helper Functions:
- IMG SIZE = (256, 256)
- Batch size = 8
- Epoch = 60

Image preprocessing
- Normalizes pixel values to [0, 1] divided by 255 for strong effects on brightness augmentations

Mask preprocessing
- Divides by 255 → mask values are soft (0.0 → 0.1) to allow non-binary labels

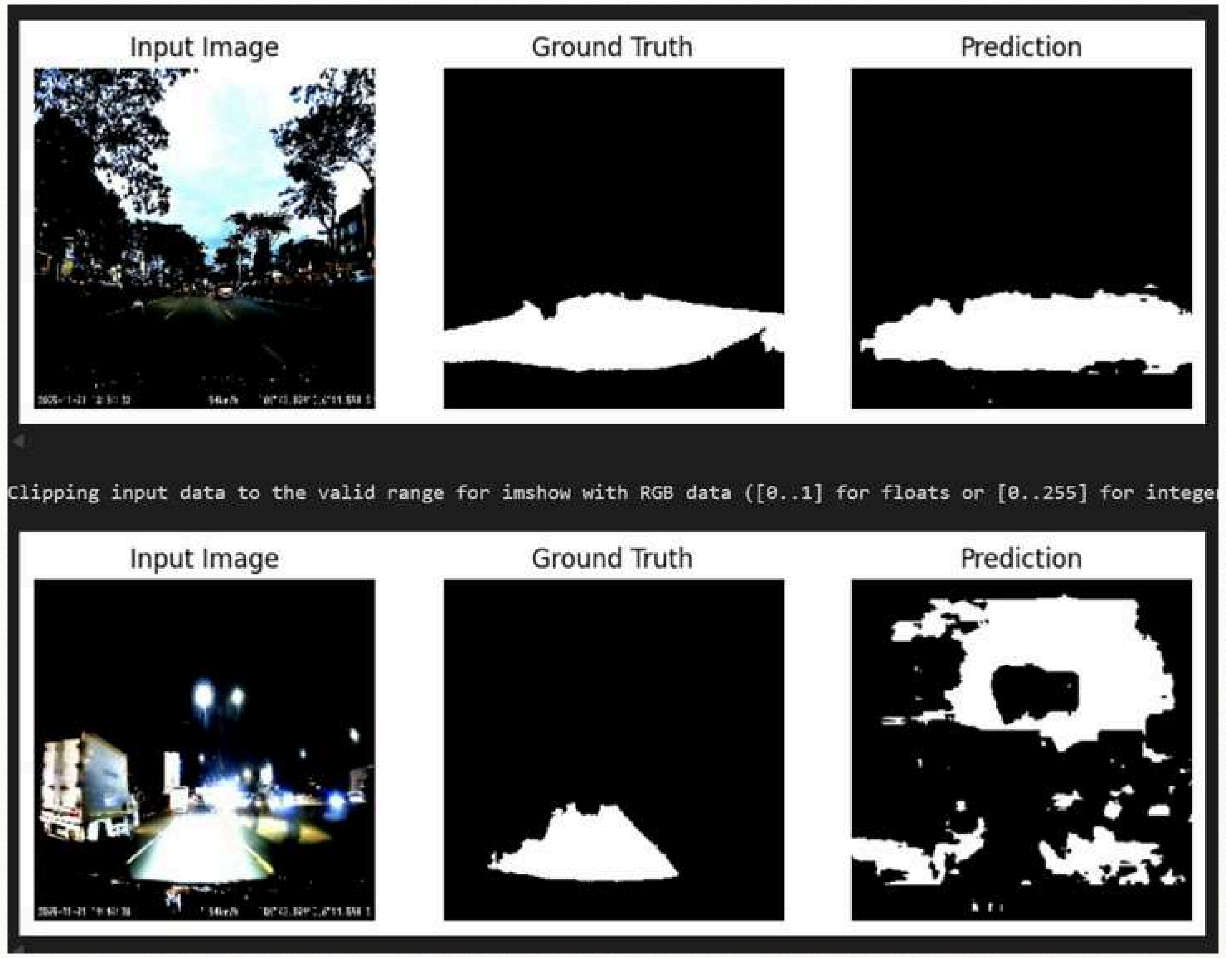
PRELIMINARY DESIGNS

V2 (NEW PARAMETERS & PREPROCESSING)



BDD100K 100IMG Dataset

- Accuracy = 95.09%
- IoU Metric = 68.53%
- Val Accuracy = 93.38%



BDD100K+Dashcam 700IMG Dataset

- Accuracy = 97.34%
- IoU Metric = 78.49%
- Val Accuracy = 83.67%

Helper Functions:

- IMG SIZE = (256, 256)
- Batch size = 16
- Epoch = 50

Image preprocessing

- Normalizes pixel values, then applies *per_image_standardization* to remove global brightness and contrast between imgs.

(*Focuses on shape/textures of the road, reduces influence of lighting changes*)

Mask preprocessing

- Normalizes pixel values and then rounds them up to enforce strict binary masks (0 or 1)

PRELIMINARY DESIGNS

SUMMARY

Aspect	**Training Model 1 (Original)**	**Training Model 2 (New)**
Image scaling	Divides pixel values by 255	Divides pixel values by 255 + Standardization
Image distribution	Original brightness preserved	Data centered at Zero with consistent spread
Brightness augmentation impact	Strong	Weak
Mask values	Soft non-binary labels	Strict binary labels
Label noise control	Lower	Higher

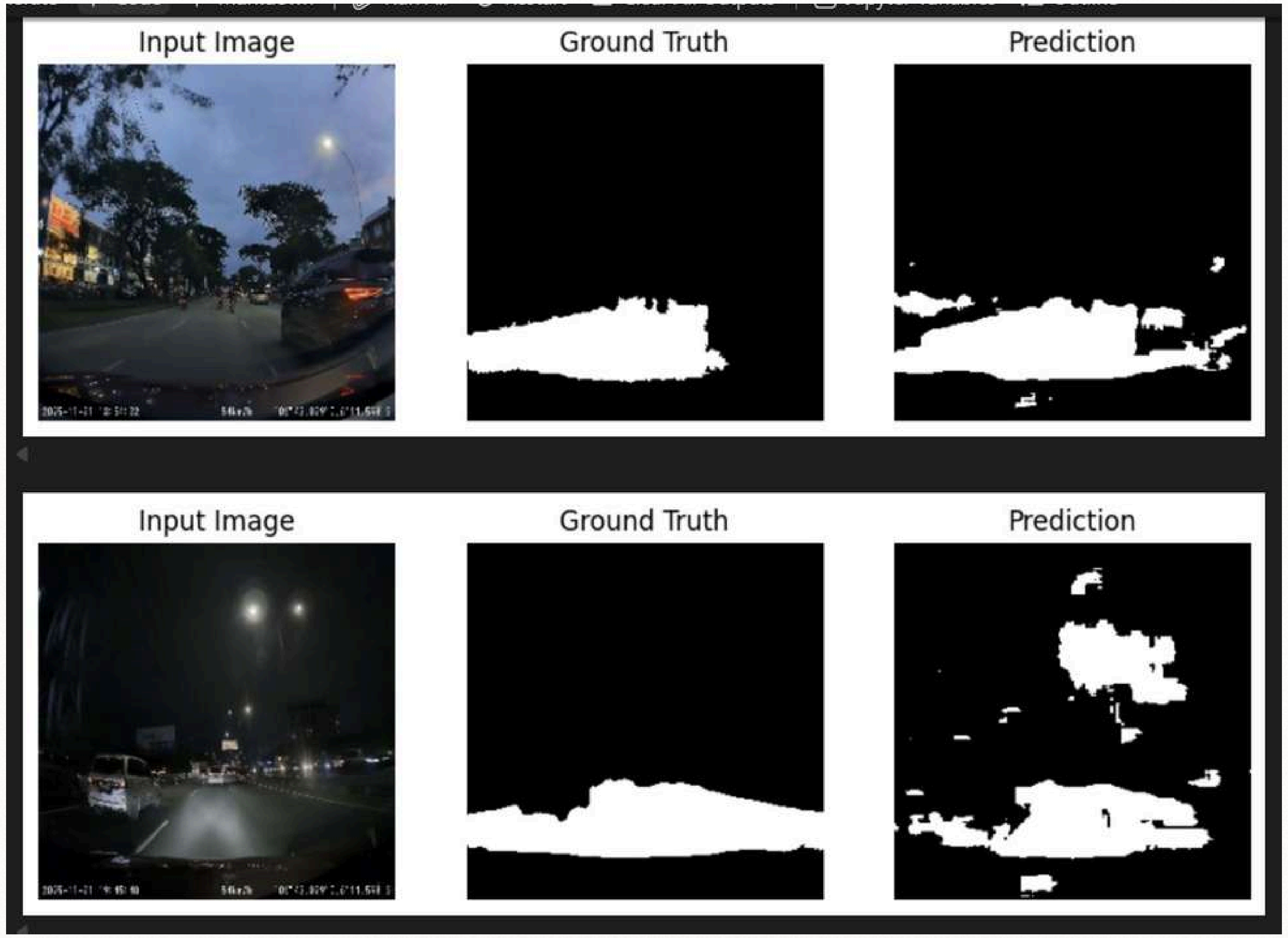
Training Model 1 (V1):
Keeps data closer to the original images and masks

Training Model 2 (V2):
Enforces stricter normalization and cleaner labels, but reduces lighting variation

FINAL DESIGN

Training Model 1 (V1)

We chose to use V1, because it keeps brightness augmentation fully effective that helps the model to learn real-world lighting variations.



BDD100K+Dashcam 700IMG Dataset

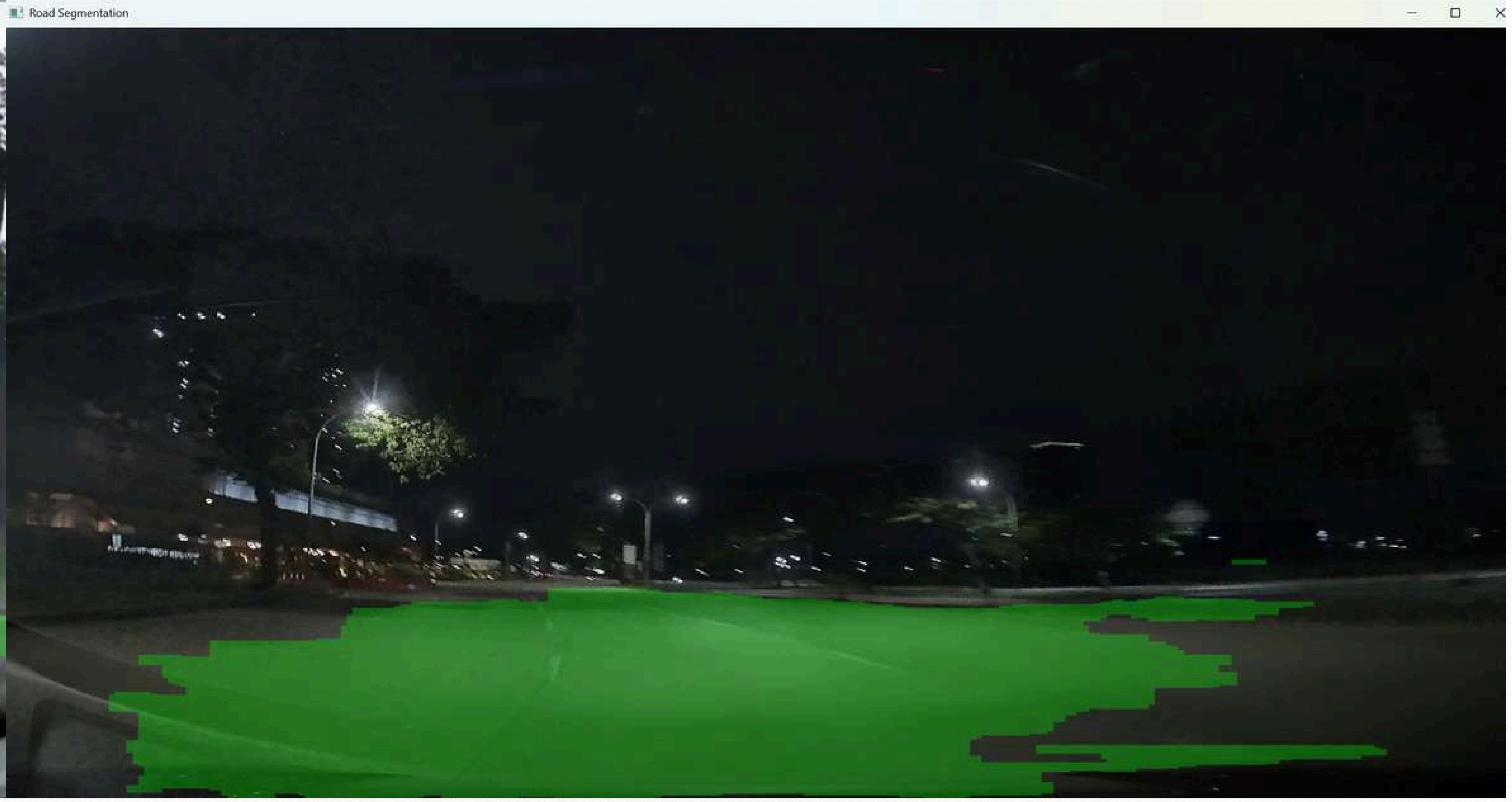
- Accuracy = **96.82%**
- IoU Metric = **73.96%**
- Val Accuracy = **94.09%**

FINAL DESIGN

Results

Day time

Night time



The system is reasonably usable for a moderate-level road area estimation for drivable area detections and lane-region approximation.

TESTING MODEL PIPELINE

1
Load Model & Setup
Define segmentation metrics (IoU) to load the trained model correctly

2
Capture Camera Frames
Continously captures video frames in real time for processing

3
ROI (Region of Interest)
Focuses the model on the road area to reduce unnecessary computations.

4
Preprocessing
ROI resized to 256x256 (match model's input)
Pixel values are normalized to [0, 1]

5
Visualization
Segmentation mask is resized back to original frame size. The detected road area is overlaid in green occupancy grid on the original frame, displayed in real-time.

CONCLUSION

The project demonstrates a working semantic road segmentation system that operates in real time using camera input. The results confirms that the trained U-Net based model is capable of identifying drivable roads under both daytime and nighttime conditions, which aligns with the original objective of acheiving road perception in varying lighting environments.

FUTURE IMPROVEMENTS

This project leaves significant room for further improvement. Common approaches to enhancing the semantic road segmentation model include increasing the size and diversity of the dataset, optimizing model parameters, and experimenting with pre-trained models such as YOLOv8 or YOLOv11 as references. However, they are a matter of experimentations of trial and error. Due to time constraints, the current implementation represents the best result achievable within the scope of this project.

THANKYOU