#### Submission date : 12/4/2023

# Improvements from previous deliverable

## Road segmentation post processing

Previously, the homography results from YOLOP were showing a poorer result. There were many holes in the segmentation, which caused a poorer overall result in the end.

Morphological operations applied were used for removing small artefacts and holes in the image using the Opening and Closing operations.

### Previous result:



Current result:



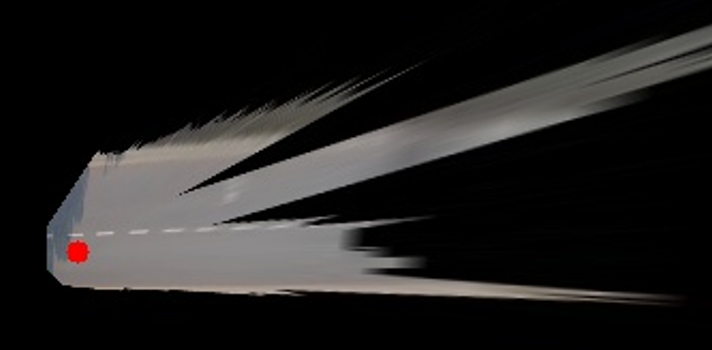
## ROI for corresponding points:

##### Points which are outside the drivable area were not considered as candidates for interframe homography calculation. This improves homography results as full scene has objects moving in different directions.

# Object detection

Implemented algorithm:

1. For the current frame get all objects detected by YOLOP
2. Find the object with highest confidence score
3. Store the coordinates of the object in JSON file format
4. While calculating projections, pick the object coordinates for each frame
5. Apply homography which is already calculated to the object point
6. Save the transformed object location in another JSON file
7. While applying projections to an image, draw a circle to the transformed object location to show object location in top view



Files edited/added:

Changes in “roadSegmentation.py”:

Since it already contained the functionality and the model, we just added the object detection feature in this file and wrote the incoming data in a new .json file ‘objects.json’.

Features of “objects.json”:

The objects file contains the bounding information in the form x ,y, confidence, and class.

x = location “x2” of bounding box

y = difference “y2 – y1”

confidence = the confidence of our selected object/vehicle

class = the label of the object(optional)

# General overview for all tasks

# GitHub Repository

“gh repo clone rizahmad/Road2Sat”

## Directory structure

The repository contains 3 main folders:

### Dataset

This folder contains all the raw data:

#### Frames:

Contains all the frames of the video being used.

#### rs\_frames:

contains the segmented lane image for each given frame.

#### roadref:

contains the initial reference frame of the ground video.

#### satref:

contains the initial reference frame for the satellite image.

### gen

this contains all the generated data and .json files containing all the interframe homographies and the initial satellite image to 1st frame homography.

### Resources

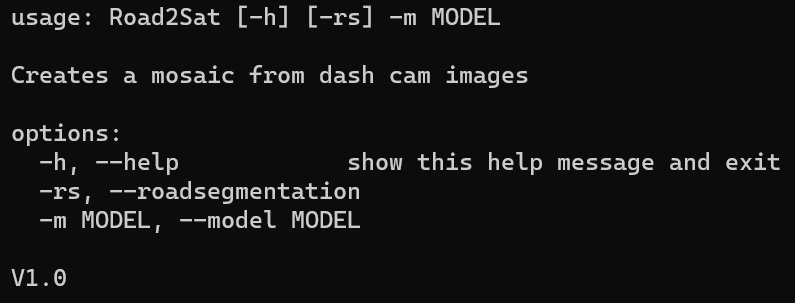
Contains the required models like superglue and YOLOP being used to generate the top-view image and the scripts folder which contains all the scripts being utilized in our image generation.

# Scripts

## Road2sat.py

Command





### Design

Calculates inter-frame homography and generates projection images and mosaic.

A close-up of a jet

Description automatically generated

## Video2frame.py

### Command

### Design

The task of this script is to store the individual frames of a video in a folder so they can be utilized later.

### Sample outputs



A road with palm trees and buildings

Description automatically generated

## roadSegmentation.py

### Command



### Design

The file utilizes the YOLOP lane segmentation to provide the final lanes(after being segmented) which are multiplied with their mask and only the lane remains.

### Sample outputs

A road with a black background

Description automatically generated

A road with a black background

Description automatically generated

## numpyArrayEncoder.py

### Command

N/A – internal script

### Design

Utility script that converts any desired numpy array into json file

### Sample outputs

Not applicable

## selectedPointsHomography.py

### Command

### Design

We use this file to hand annotate the corresponding points between the satellite image and 1st frame and compute the homography using cv2.

### Sample outputs (computed homography)

