**Project Title:** Road Condition Detection

**Team members:**

Phuc Le, [Phuc.le01@sjsu.edu](mailto:Phuc.le01@sjsu.edu), 013873028

Phuoc le, [phuoc.le@sjsu.edu](mailto:phuoc.le@sjsu.edu), 013868400

Tommy Dao, [tommy.dao@sjsu.edu](mailto:tommy.dao@sjsu.edu), 013958178

**Proposed project:** Default project 1: AI Object Detection for Smart City

**Dataset:**

* COCO (Common Objects in Context): <https://cocodataset.org/>
  + Dataset labelling common objects such as cars, streetlights, people, trash, etc.
* BDD100K (A Large-scale Diverse Driving Video Database): <https://bair.berkeley.edu/blog/2018/05/30/bdd/>
  + Video database of common objects on the road
* RDD2022 (The Multi-National Road Damage Dataset 2022): <https://datasetninja.com/road-damage-detector>
  + Video database of road conditions such as potholes and road cracks

**Final Demo:**

For the final demo, we aim to deliver a model that can detect different obstacles that may interfere with traffic conditions. These obstacles include common objects such as pedestrians, trash and other common objects. When detecting people or vehicles, we will make it so that faces and license plates are blurred for privacy reasons. The model will need to be able to detect a person’s face from identifying a person and also the license plate within the car. We hope to also include uncommon objects such as road cracks and potholes which also contribute to road congestion. Our model should be able to detect these different conditions through a variety of videos, labelling each obstacle accordingly within the video.

**Reference**:

Detectron2: <https://github.com/facebookresearch/detectron2?tab=readme-ov-file>

YOLO8: <https://github.com/ultralytics/ultralytics>

# **Project Plan:**

* Project’s goal
  + Create a web app where users can upload videos and it will output a visualization of the objects with labels
* Modify Yolov8 model to detect
  + Use Roboflow to convert data format to yolov8 format
  + Road cracks (U.S. dataset): <https://datasetninja.com/road-damage-detector>
  + Road signs, people, and cars: <https://cocodataset.org/>
  + License plate, trash, and plants (need data)
* Two options to train model
  + Professor’s code: DeepDataMiningLearning - <https://github.com/lkk688/DeepDataMiningLearning/tree/main/DeepDataMiningLearning>
  + Ultralytics’ github repo: <https://github.com/ultralytics/ultralytics>
* Plans of architecture change
  + Use different models scaling such as yolov8m, yolov8l, volov8x
  + Replace backbone layers with MobileNetV2 or ConvNeXt
  + Potentially add or remove layers, try a squeeze-and-excitation block
  + Change activation functions(ReLu, LeakyReLU, GELU)
  + Add dropout

# **What We Have Done So Far:**

* We are planning to train using yolov8n from Ultralytics
* Used roboflow to convert the format of the Road cracks dataset to the yolov8 data format
* Testing if the yolov8n model is able to train with our current dataset and custom architecture.
* Test trained a yolov8n model with a rock paper scissor dataset to test training of custom architecture via the yolov8.yaml file.
  + We learned that it takes a lot of data to train a yolov8n model to have good performance, and since we don't have a lot of data, we tried transfer learning from the weights of the pre-trained yolov8n.pt model.
  + The only downside of transfer learning is that if the custom model is too different from the pretrained model in terms of architecture, then only some layer’s weight will be able to transfer to the custom model.
  + Google Colab: <https://colab.research.google.com/drive/1n2ARLQywlOW3cuYHte33As6V9vWVU21V?usp=sharing>

# **Current Roadblock:**

* Formatting datasets to be compatible with Yolov8
* Trying to train a model with a small amount of data to test for accuracy
* Trying to determine which methods of training/validation will be more efficient