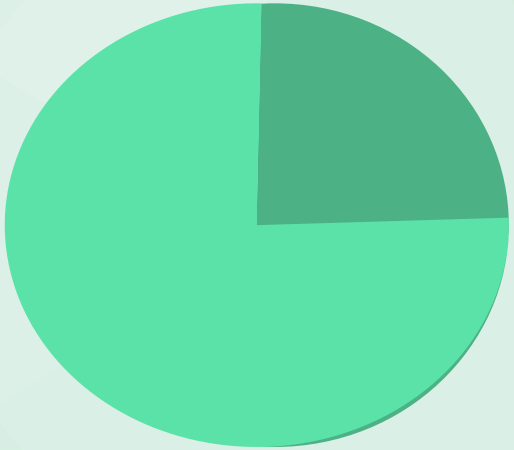
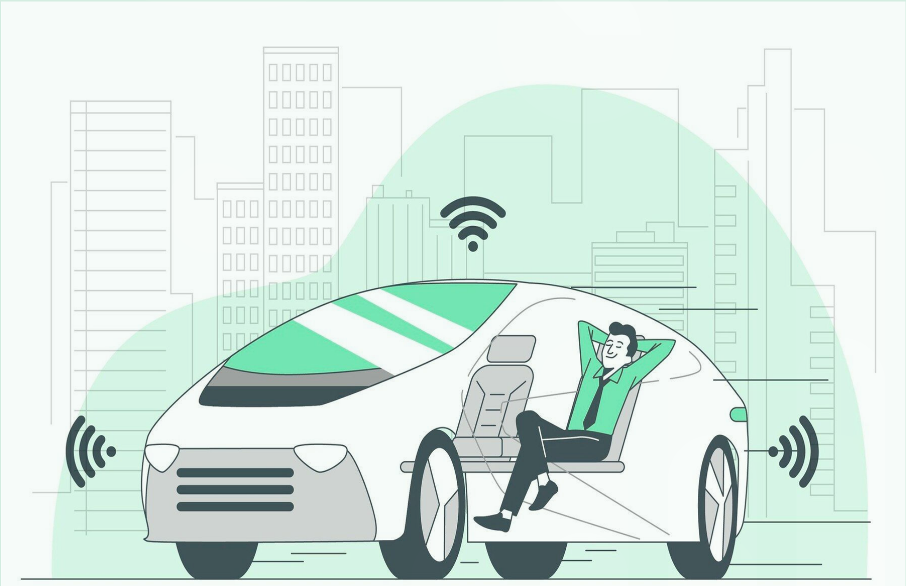


Deep Learning and Computer Vision Algorithms for Self-Driving

Shilpi Shah, Brendan Franz, Travis Forgach, Milan Jostes

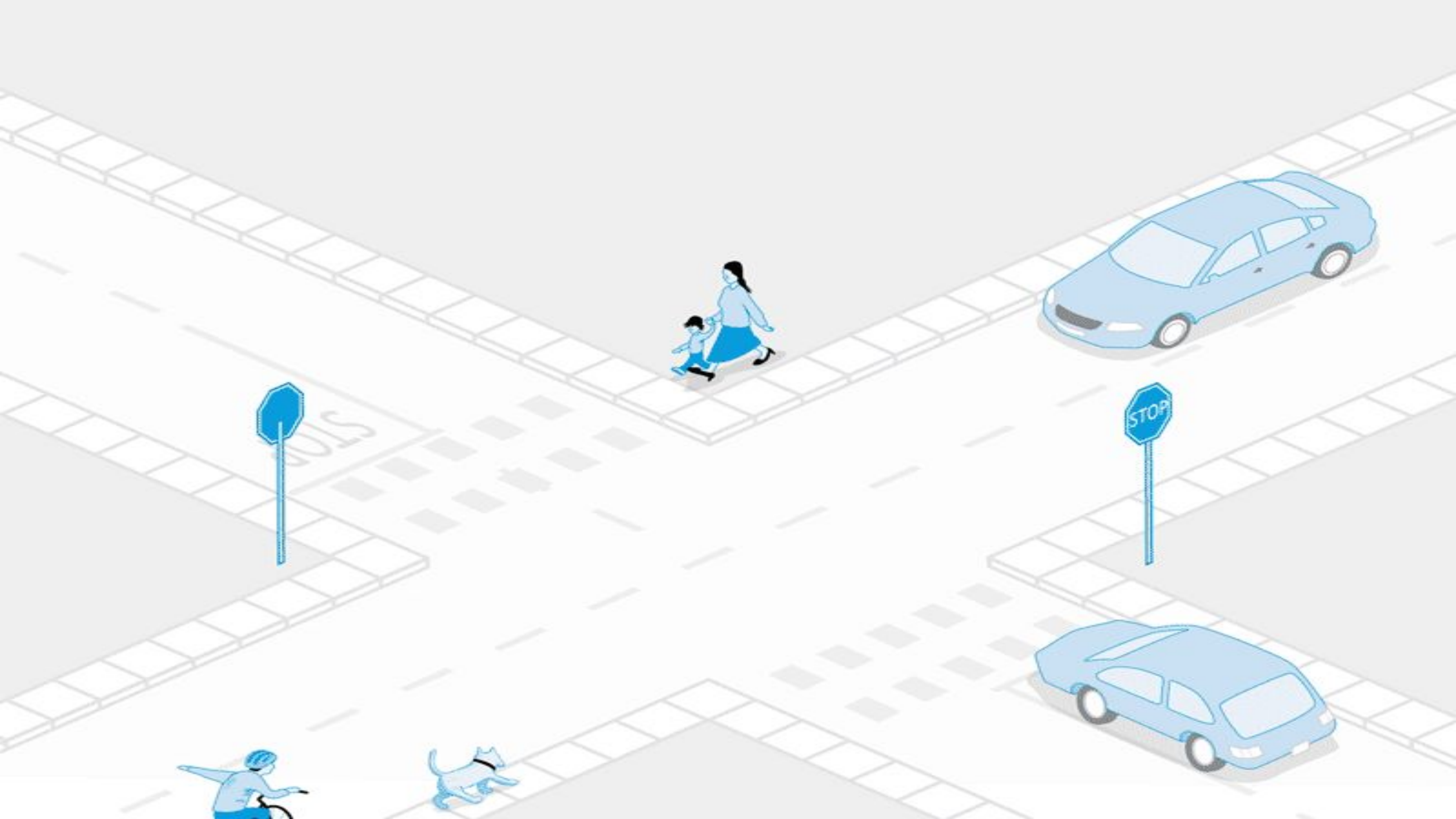


Supported through NSF grant 2150292 and 2150096

OVERVIEW

1. Motivation
2. Background
3. Objective and Goals
4. Experimental Setup
5. Methodology: Traditional
6. Methodology: Deep Learning
7. Future Work

MOTIVATION


STOP

Incentives for Self Driving Cars

- Increased Safety
- Increased Productivity
- Increased Efficiency
- Integration with smart cities

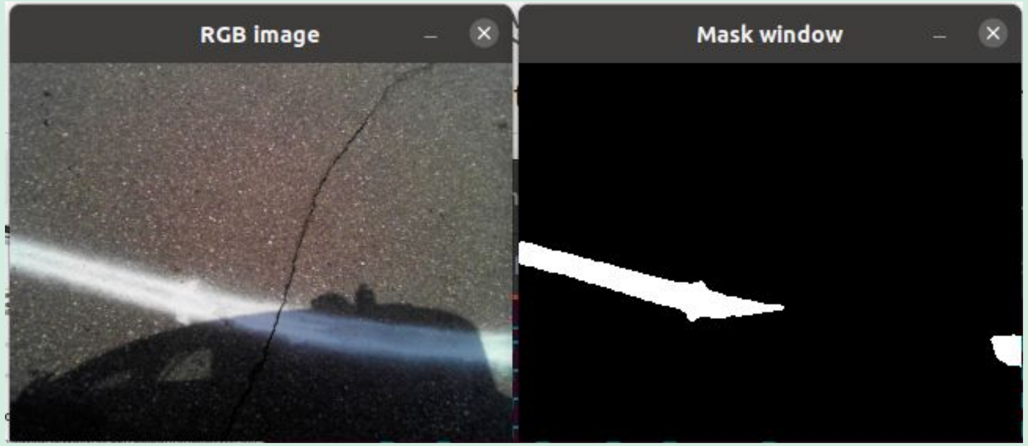
BACKGROUND

Previous Work

- Computer Vision Algorithms
- Convolutional Neural Network Algorithms

Objective and Goals

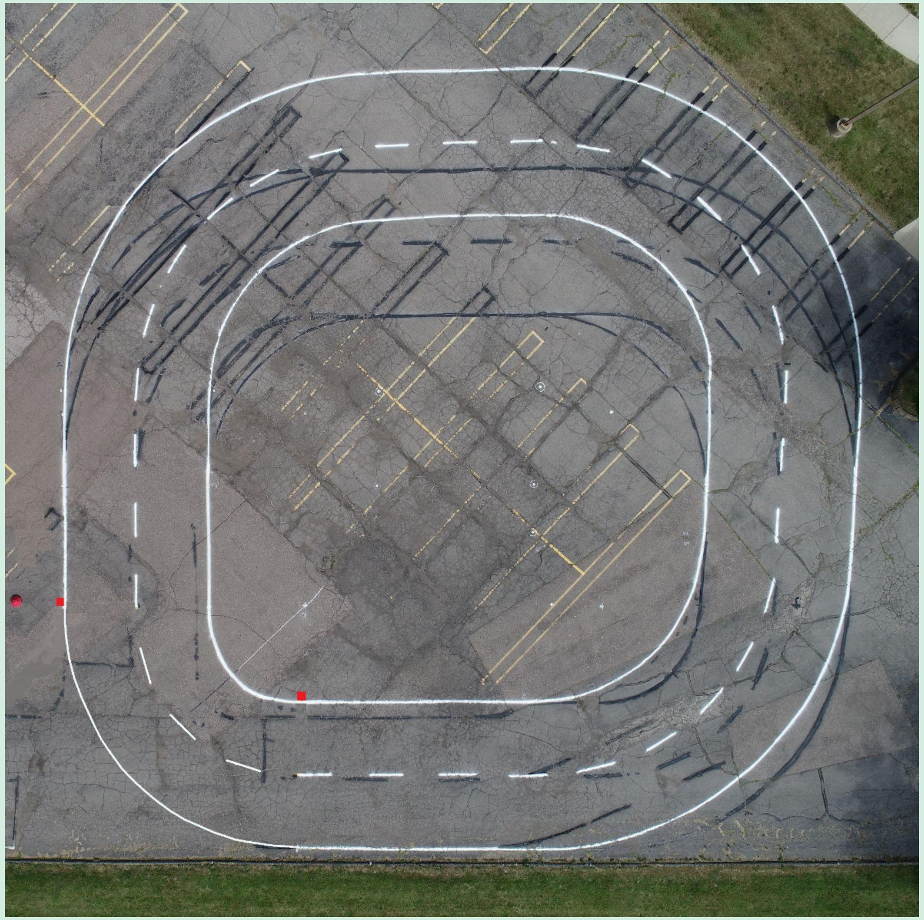
Performance in Adverse Conditions



- Shadows
- Sharp Curves
- Unclear White Lines
- Overexposing Sunlight
- Rain
- Poor Pavement Conditions
- Puddles

Experimental Design

Our Test Course



- Outer lane center length: 86.32 meters
- Outer line: 106.97 meters
- Inner lane center length: 71.43 meters
- Inner line: 69.27 meters
- Red squares on the track to act as stop signs

Our Test Car

- For our project we used LTU’s ACTor Vehicle
    - Modified Polaris Gem e2 electric vehicles
    - Drive-by-wire system
    - Mako camera
    - Ubuntu linux computers
    - ROS and Python

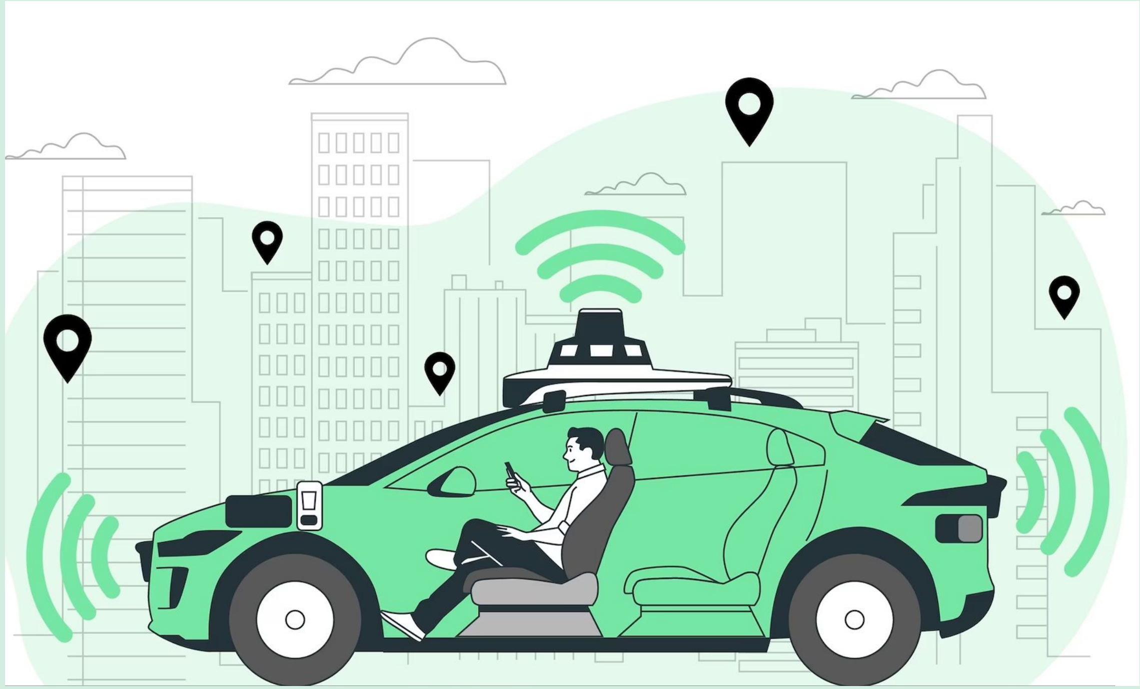
Demonstration Evaluation

- Target number of laps: 5
- Lane (circle one): Inner Outer
- Date/Time of the run:
- Weather condition: Sunny, partly-cloudy, completely cloudy, light rain, heavy rain

Laps Completed per Algorithm	Deep Learning	Time Taken	Computer Vision	Time Taken
Inner	__	___	__	___
Outer	__	___	__	___

METHODOLOGY: TRADITIONAL

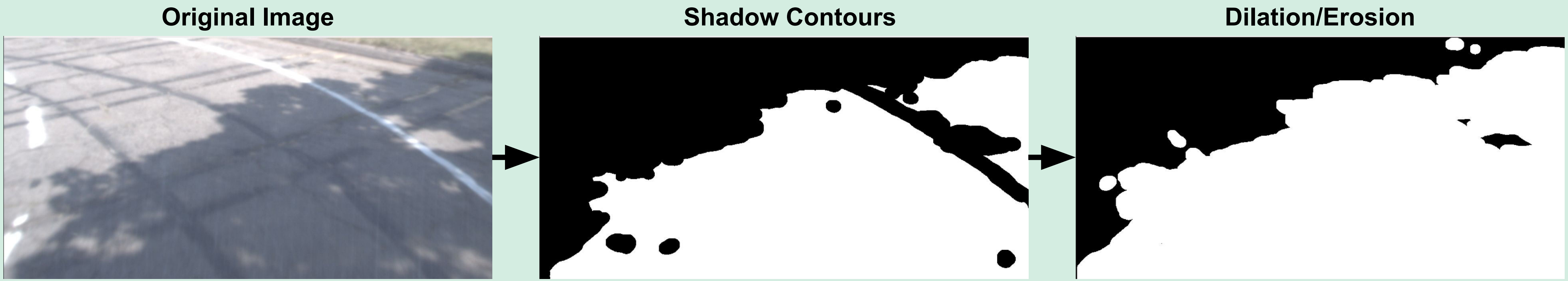
Traditional Algorithm Self-Driving



- Create a baseline program for self-driving
- Identify areas of improvement and limitations of traditional algorithms

Preprocessing

- Identify and remove yellow lines
- Identify shaded areas and apply fitting functions
- Dilate and erode shaded contours for increased deadzone

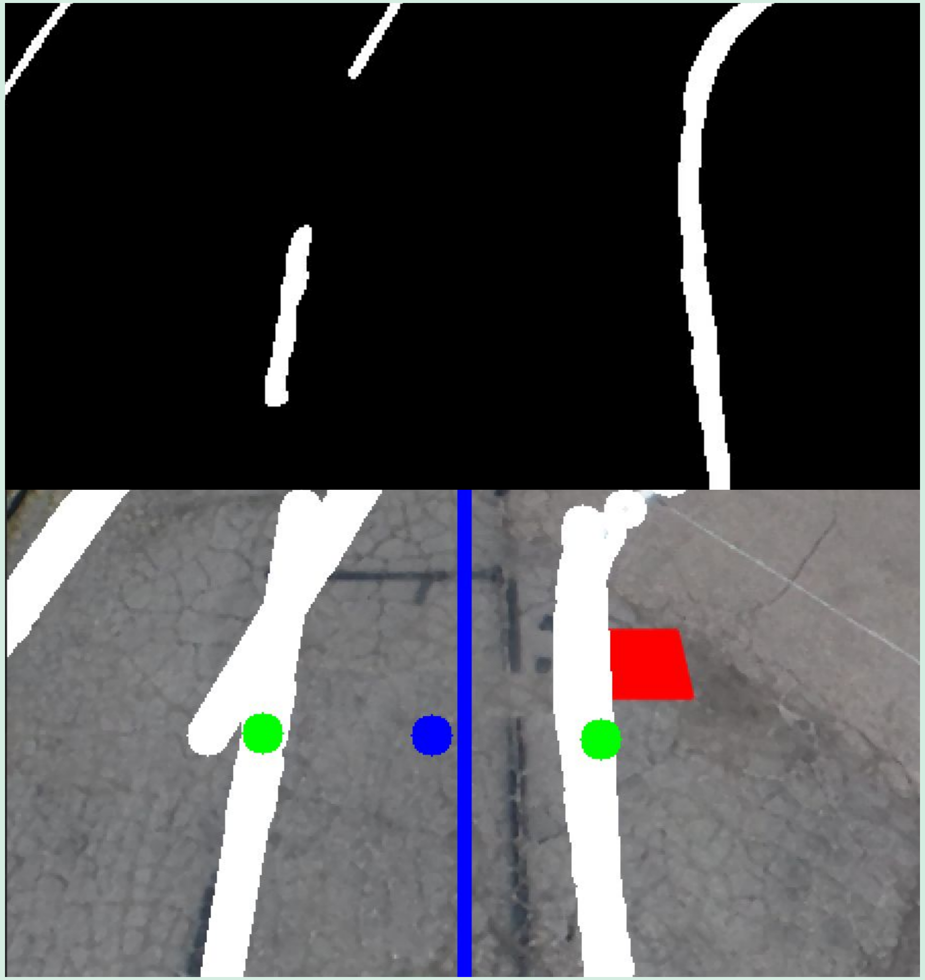


Preprocessing

- Increase brightness of shaded areas
- Adjust contrast to regulate white line intensity
- Find new contours and add them to lane contours

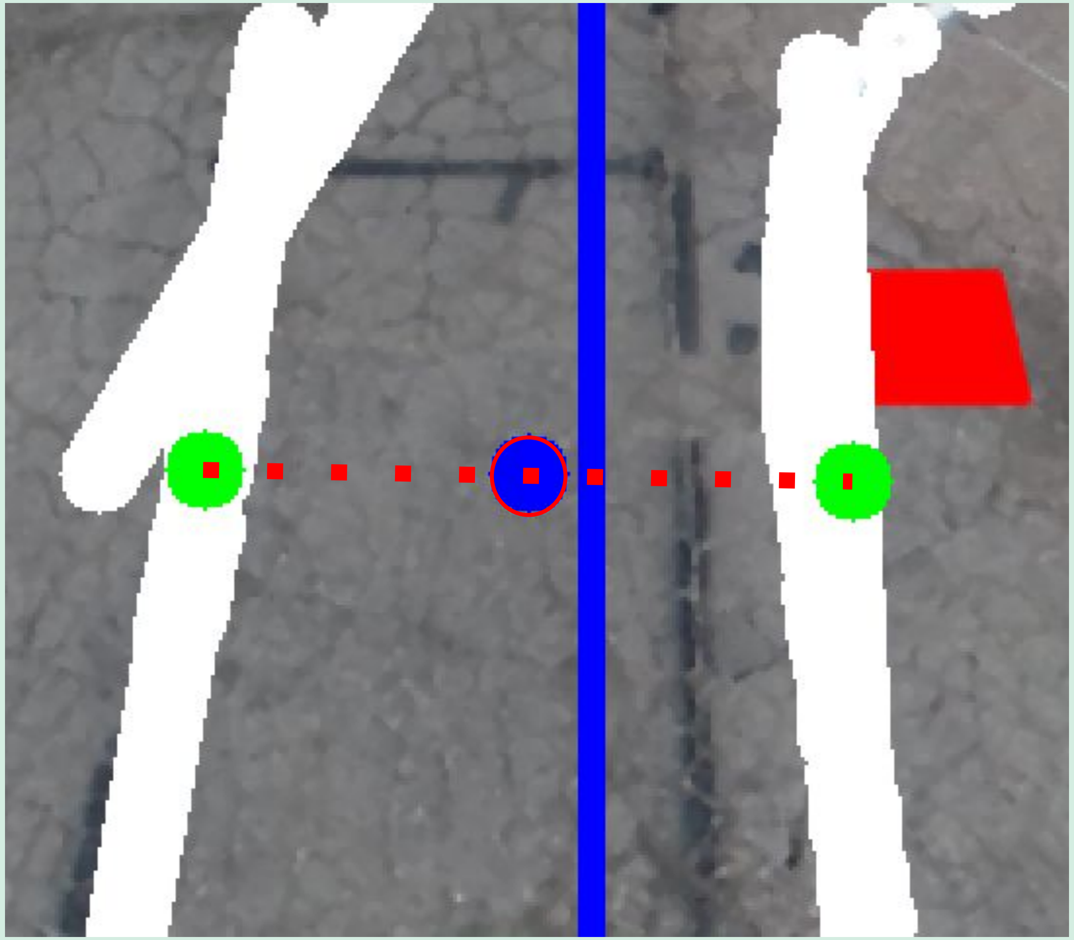
Apply Brightness Mask **Adjust Contrast** **Find Shadow Contours**

Lane Following



- Identify possible lane contours
  - Dilate contours to fill gaps
- Take five largest contours and extend them
  - Helps identify dashed line
  - Connect close contours to form one line

Lane Following

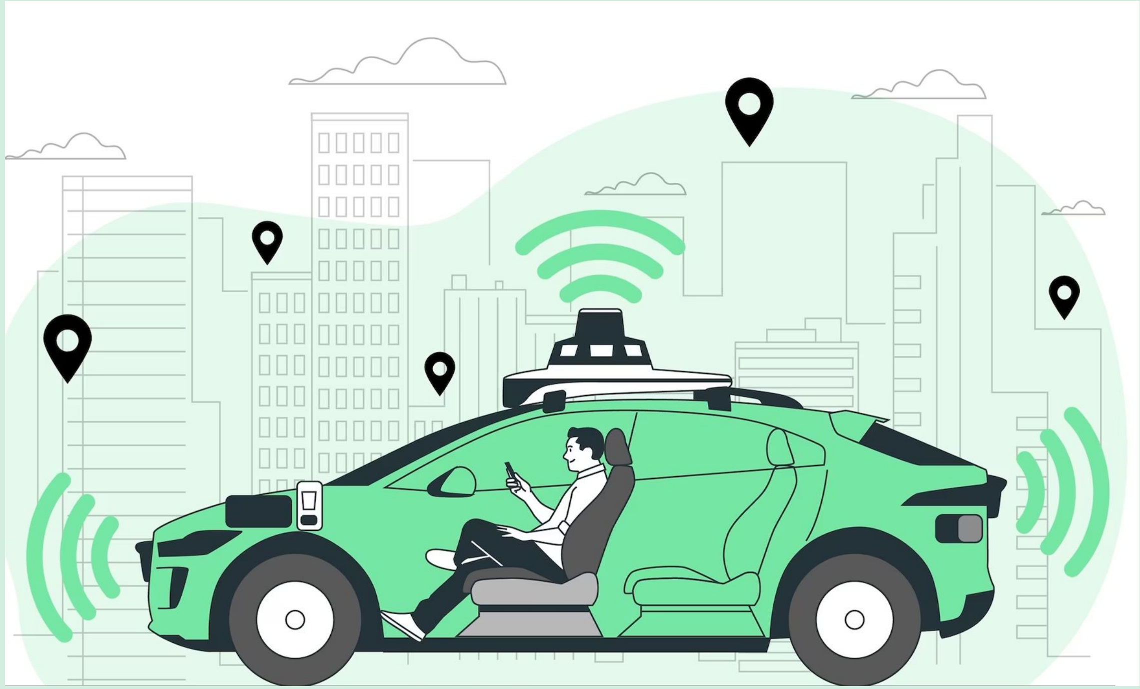


- Identify most probable lines
  - Based on previous points
- Find the center point of the lines
  - Use this to determine center of lane
- Align center of vehicle with the center of the lane

Current Problems

- Inner lane loses sight of the inner line on tight corners
- Sudden extreme changes in lighting conditions can cause false line identification

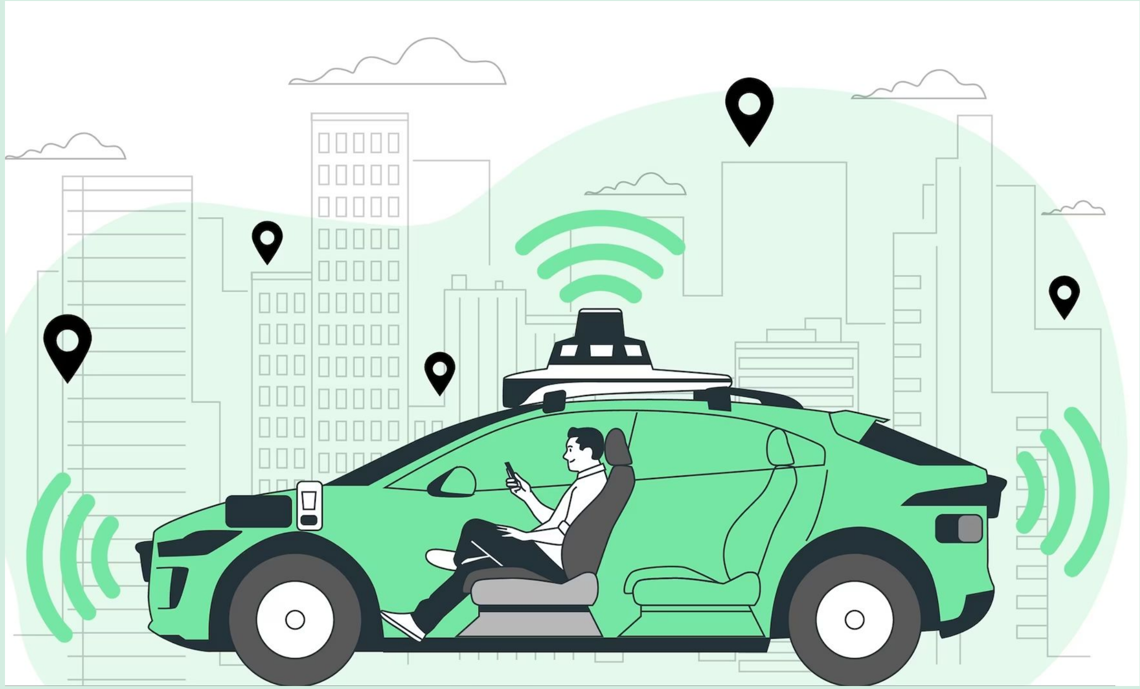
Future Solutions



- Wide lens camera
    - Turn Prediction
- Improved preprocessing speed
    - Improve dynamic lighting adjustment
    - Stop when lighting changes drastically

METHODOLOGY: DEEP LEARNING

Project DeepSteer



- Overcome the limitations of traditional computer vision algorithms
- Adapt to weather and lighting conditions

Convolutional Neural Network

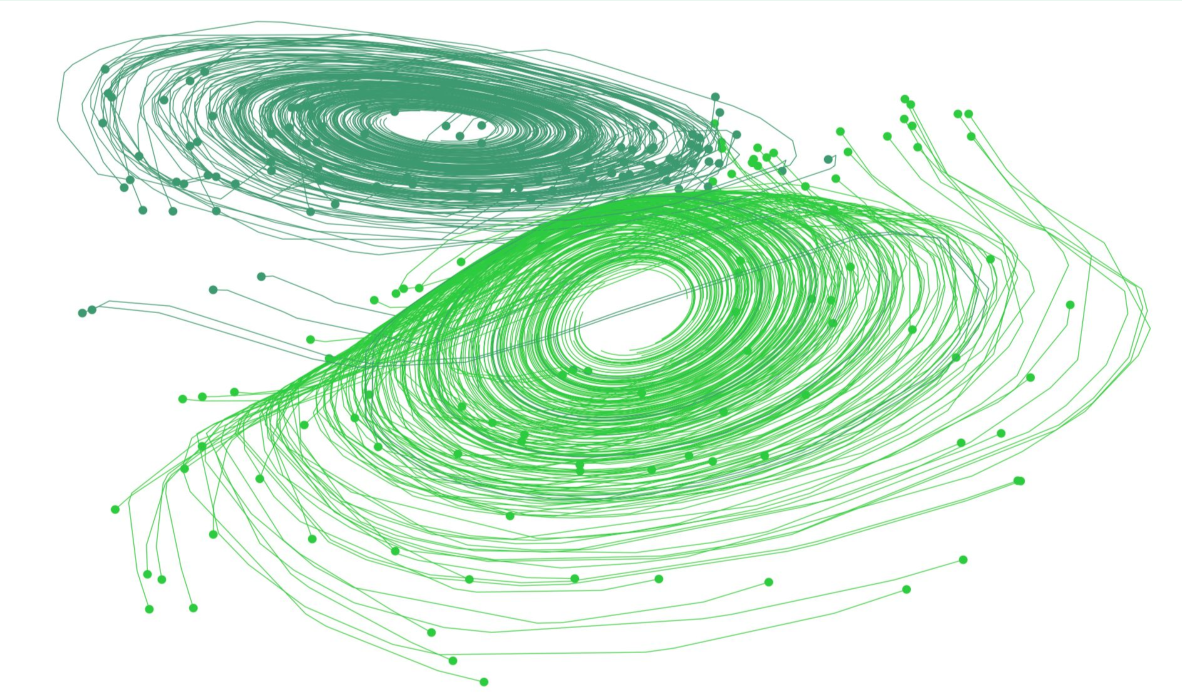
- Sequential
- Convolutional, Dense, Max Pooling and Flatten layers

```python
model = tf.keras.Sequential([
    layers.Conv2D(32, (3, 3), activation=tf.nn.leaky_relu, input_shape=(100, 100, 3)),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation=tf.nn.leaky_relu),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation=tf.nn.leaky_relu),
    layers.Flatten(),
    layers.Dense(64, activation=tf.nn.leaky_relu),
    layers.Dense(1)
])

opt = tf.keras.optimizers.SGD(learning_rate=0.001)

model.compile(optimizer=opt, loss="mse", metrics=["mae"])
```

Other Types of Neural Networks



- Recurrent Neural Networks (RNN)
  - Long Short-Term Memory Networks (LSTM)
- Recurrent Convolutional Neural Networks (RCNN)

Data Collection with ROS

- ROS bag collection in all weather conditions
- Subscription to Drive By Wire and Camera topics
- Extraction of bags and dataset creation

APIs

- TensorFlow
  - Keras
    - Model
      - batch = 1
      - epochs = 10
- scikit learn

Preprocessing

- Images and velocity Twist messages
- Resizing
- End to end imaging
- Grey-scale images

Training and System Integration

- Extraction Program and Data Set Creation
- CNN Training
- Yaw Prediction

```
image_path,angular_z
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739830.7501805.png,-0.0
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739830.8500342.png,-0.0
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739830.9500625.png,-0.0
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739831.0500908.png,-0.0
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739831.1500397.png,-0.0
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739831.2500775.png,-0.00013371485480983755
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739831.3501673.png,-0.00018448763914725956
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739831.4499707.png,-0.00010387113706825905
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739831.5501604.png,-0.00023564759694501112
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739831.650197.png,-0.00011433531834432036
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739831.7501056.png,-0.0001360393638653124
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739831.850126.png,-6.64692408186365e-05
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739831.9501455.png,-5.697349374266326e-05
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739832.0501812.png,-1.5890561185667047e-05
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739832.1501634.png,-5.1644324852366465e-05
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739832.2501087.png,-0.0001469876893025059
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739832.350199.png,-0.00019068673955573024
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739832.450088.png,-0.000258221629589558
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739832.5502079.png,-0.0003416470581662187
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739832.6501212.png,-0.00043747519391016223
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739832.75014.png,-0.0005929913376917597
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739832.8501837.png,-0.0009226325439159054
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739832.9501927.png,-0.0011856097271350549
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739833.0500941.png,-0.0013697105418507818
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739833.1501172.png,-0.0015188314317316506
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739833.2500803.png,-0.0017223113056785537
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739833.3501647.png,-0.001802249777464037
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739833.450165.png,-0.002150298445008226
/home/reu-actor/actor_ws/src/deepsteerv1/images/images20/1688739833.5500643.png,-0.0024262617885611173
```

Statistical Results

- Loss
- MAE
- Predicted Yaw Rate

```
Successfully loaded the model.
Epoch 1/10
1443/1443 [==============================] - 11s 7ms/step - loss: 0.0014 - mae: 0.0299
Epoch 2/10
1443/1443 [==============================] - 10s 7ms/step - loss: 0.0014 - mae: 0.0297
Epoch 3/10
1443/1443 [==============================] - 10s 7ms/step - loss: 0.0014 - mae: 0.0294
Epoch 4/10
1443/1443 [==============================] - 10s 7ms/step - loss: 0.0014 - mae: 0.0292
Epoch 5/10
1443/1443 [==============================] - 10s 7ms/step - loss: 0.0014 - mae: 0.0291
Epoch 6/10
1443/1443 [==============================] - 10s 7ms/step - loss: 0.0013 - mae: 0.0291
Epoch 7/10
1443/1443 [==============================] - 10s 7ms/step - loss: 0.0013 - mae: 0.0286
Epoch 8/10
1443/1443 [==============================] - 10s 7ms/step - loss: 0.0013 - mae: 0.0285
Epoch 9/10
1443/1443 [==============================] - 10s 7ms/step - loss: 0.0013 - mae: 0.0285
Epoch 10/10
1443/1443 [==============================] - 10s 7ms/step - loss: 0.0013 - mae: 0.0282
reu-actor@ACTOR1:~/actor_ws/src/deepsteerv1$
```

Statistical Results

- Loss
- MAE
- Predicted Yaw Rate

```
1
2/2 [==============================] - 0s 27ms/step
Feature 0
Model predicted: 0.057767242193222046. Actual label: 0.019648785201798
Loss of: 0.03811845699142405
Feature 1
Model predicted: 0.047891322523355484. Actual label: 0.135571990821388
Loss of: 0.0876806682980325
Feature 2
Model predicted: 0.03142659366130829. Actual label: 0.02196802972678
Loss of: 0.009458563934528288
Feature 3
Model predicted: 0.10416354984045029. Actual label: 0.1203469056666434
Loss of: 0.01618335582619311
Feature 4
Model predicted: 0.0322708822786808. Actual label: 0.064658863998486
Loss of: 0.0323879817198052
Feature 5
Model predicted: 0.1057509258389473. Actual label: 0.1153947897929689
Loss of: 0.00964386395402l6
```

RESULTS AND FUTURE WORK

Results

Laps Completed per Algorithm	Deep Learning	Time Taken	Computer Vision	Time Taken
Inner	5	6:50	2.9	5:30
Outer	5	7:40	4.25	7:00

Future Work

- Combination of Deep Learning and Traditional Computer Vision Algorithms
- Further Data Collection for Deep Learning Algorithms in Various Conditions

THANK YOU!
QUESTIONS?

References

- https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries
- https://businesschief.com/technology-and-ai/how-autonomous-vehicles-could-improve-human-productivity
- https://onlinelibrary.wiley.com/doi/full/10.1002/rob.21918