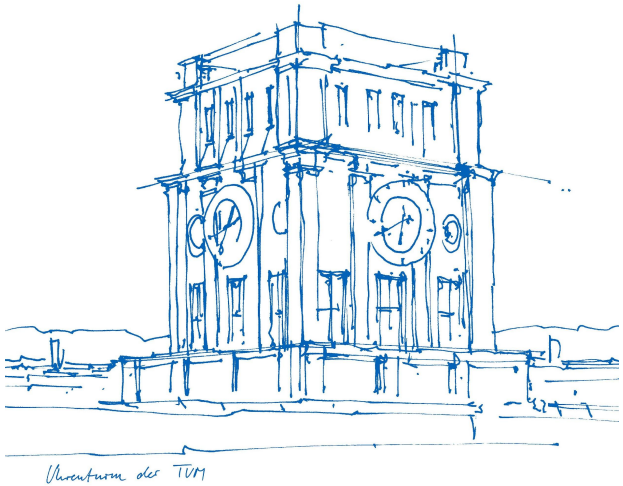
Simulation-Based Autonomous Driving in Crowded City

Final Task - Final Presentation

Ferdinand Duetsch,
Michael Sodamin

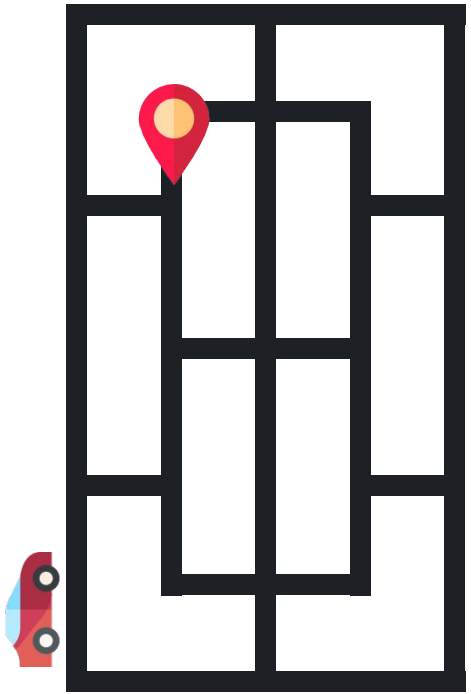
Supervisor: Liguo Zhou

Munich, 11.09.2023



Overview

1. General Approach
2. Training a Model
3. Deploying the Model
4. Evaluation
5. Demo



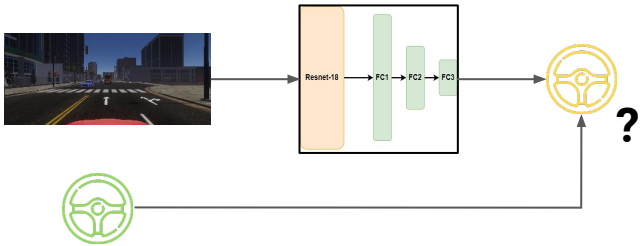
General Approach

1. Collect Dataset

Images

VehicleData

2. Train a model



Training

Deployment

General Approach

1. Collect Dataset

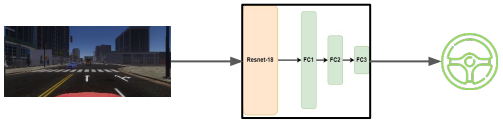
Images

VehicleData

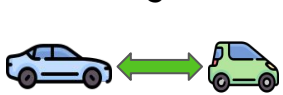
2. Train a model

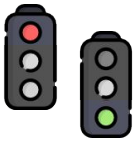
Training

1. Steering Model

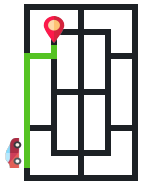


2. Detect Surroundings





3. Follow Route



Deployment

General Approach

1. Collect Dataset

Images

VehicleData

2. Train a model

Training

1. Steering Control

2. Detect Surroundings

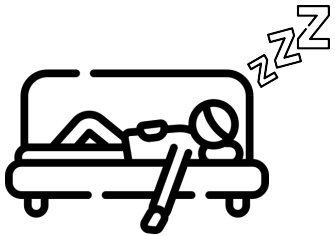
3. Follow Route

Deployment

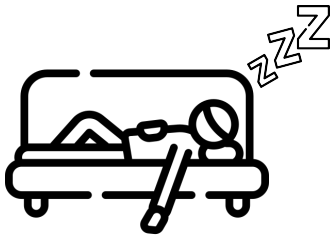
Data Collection - Manual

- Time consuming
- Mistakes in data
- Inconsistent driving behavior

Data Collection - Automated

+ No mistakes
+ Consistent driving behavior
+ Speed up simulator time
+ 

Data Collection - Automated

+ No mistakes
+ Consistent driving behavior
+ Speed up simulator time
+ 

But: Bad Performance!

Data Collection - Automated

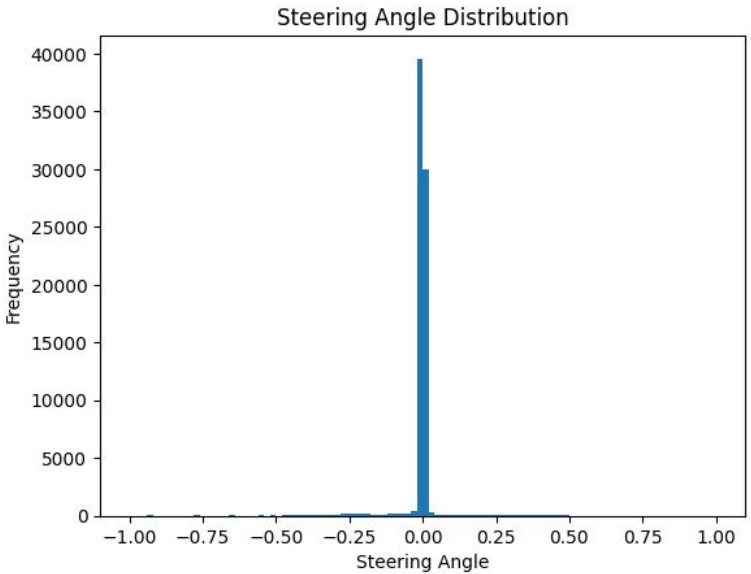
Car doesn't turn?

Recover from bad situation?

Data Collection - Balancing

Car doesn't turn?

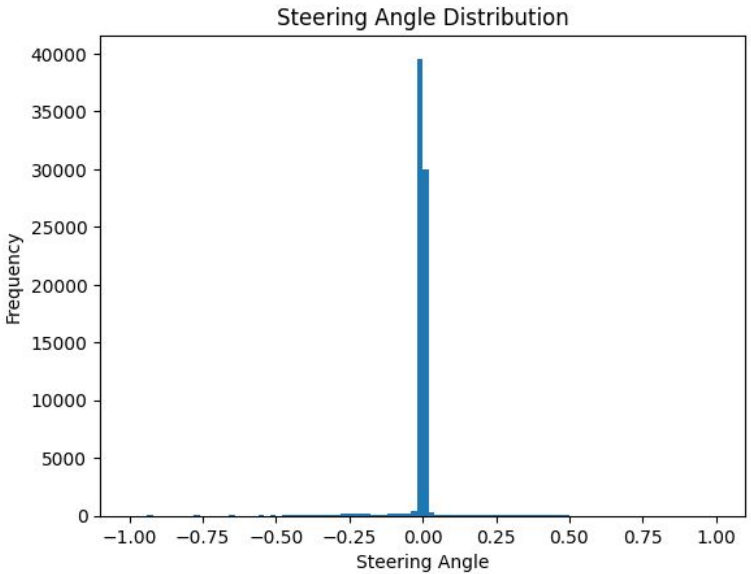
Recover from bad situation?

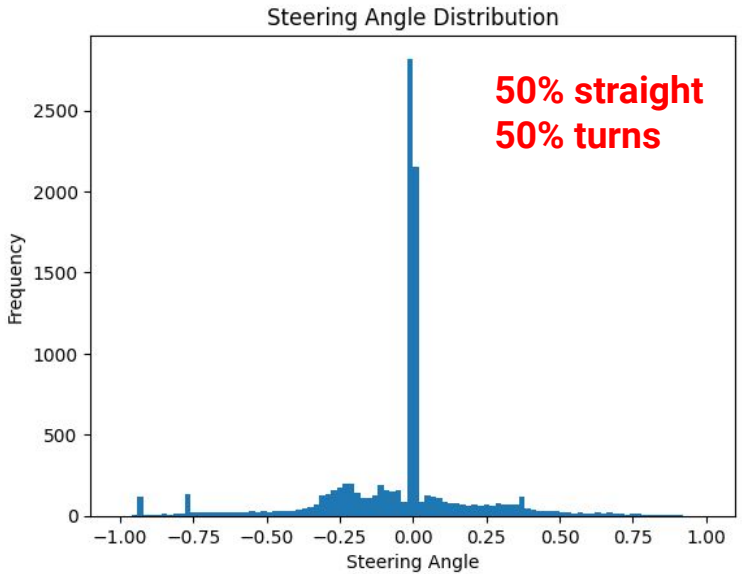


Data Collection - Balancing

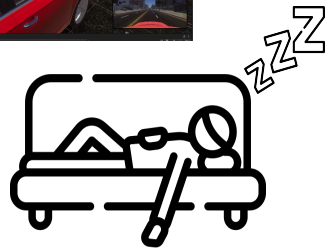
Car doesn't turn?

Recover from bad situation?





Data Collection - Hybrid Data Collection



Car doesn't turn?

Recover from bad situation?

Data Collection - Hybrid Data Collection

Car doesn't turn?

Recover from bad situation?

&

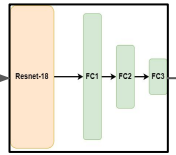
As in [1]: Expert remains in the loop, Codevilla et. al.

General Approach

1. Collect Dataset

Images VehicleData

2. Train a model



?

Training

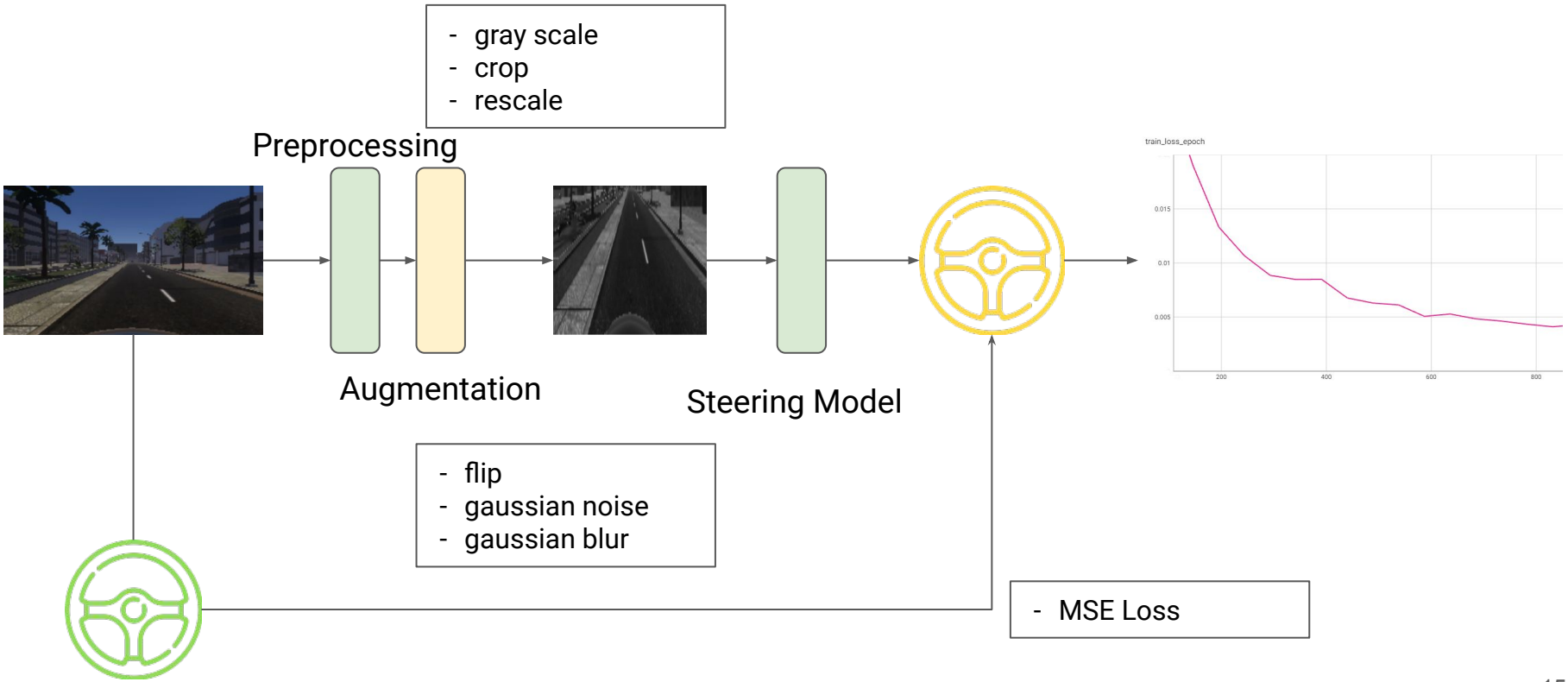
1. Steering Model

2. Detect Surroundings

3. Follow Route

Deployment

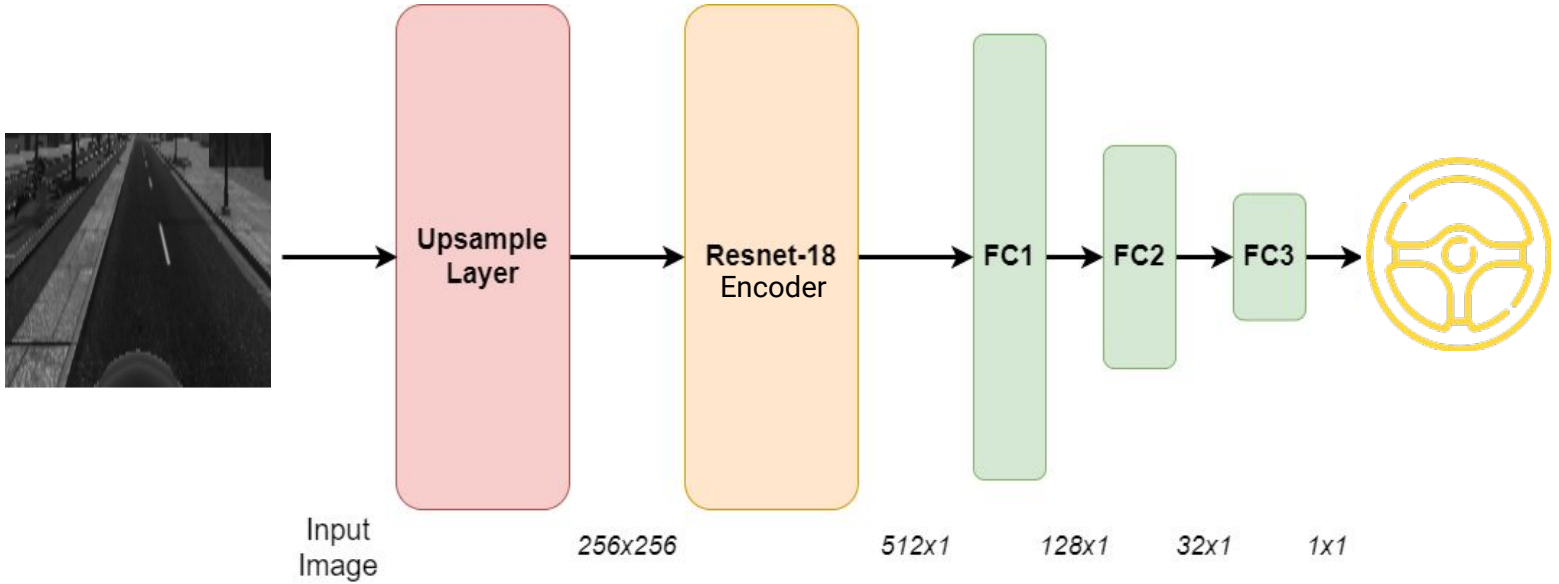
Steering Model - Training



Steering Model - Architecture

~~CNN LSTM~~

~~CSP SG~~



General Approach

1. Collect Dataset

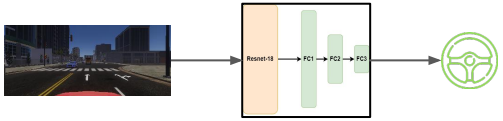
Images

VehicleData

2. Train a model

Training

1. Steering Model

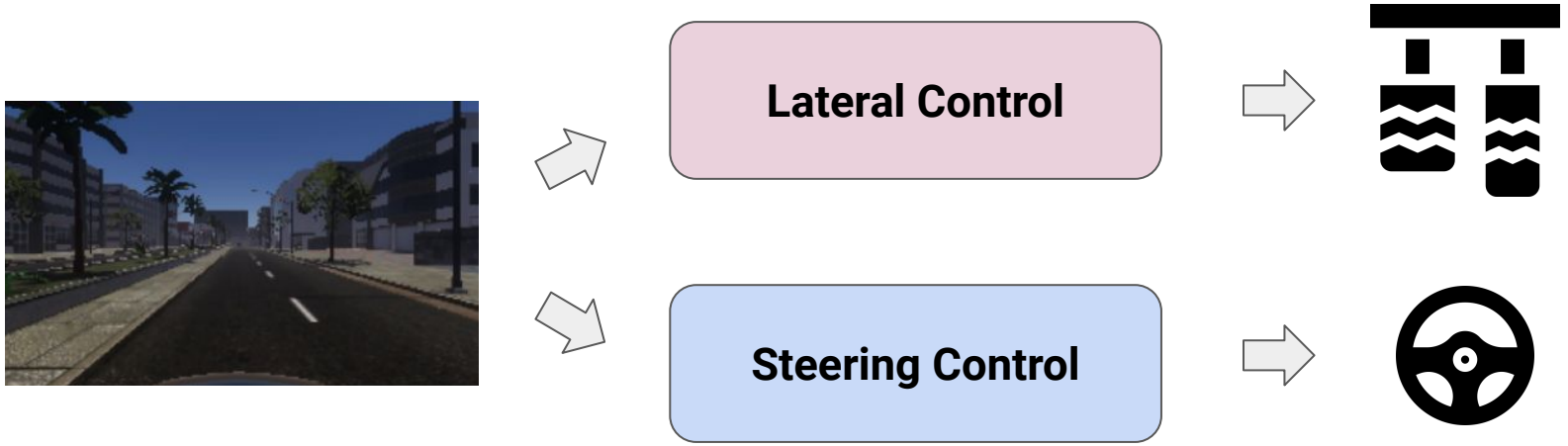


2. Detect Surroundings

3. Follow Route

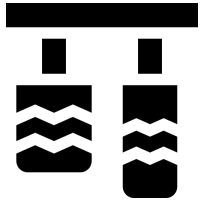
Deployment

Deployment - Software Architecture



Deployment - Software Architecture

Lateral Control

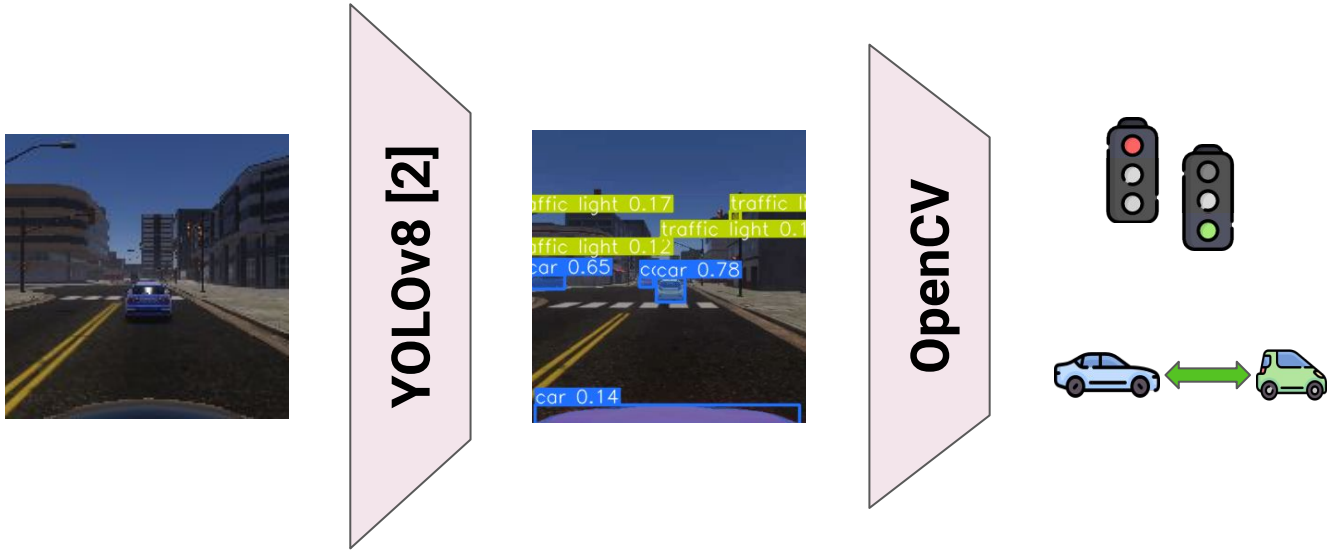


Steering Control

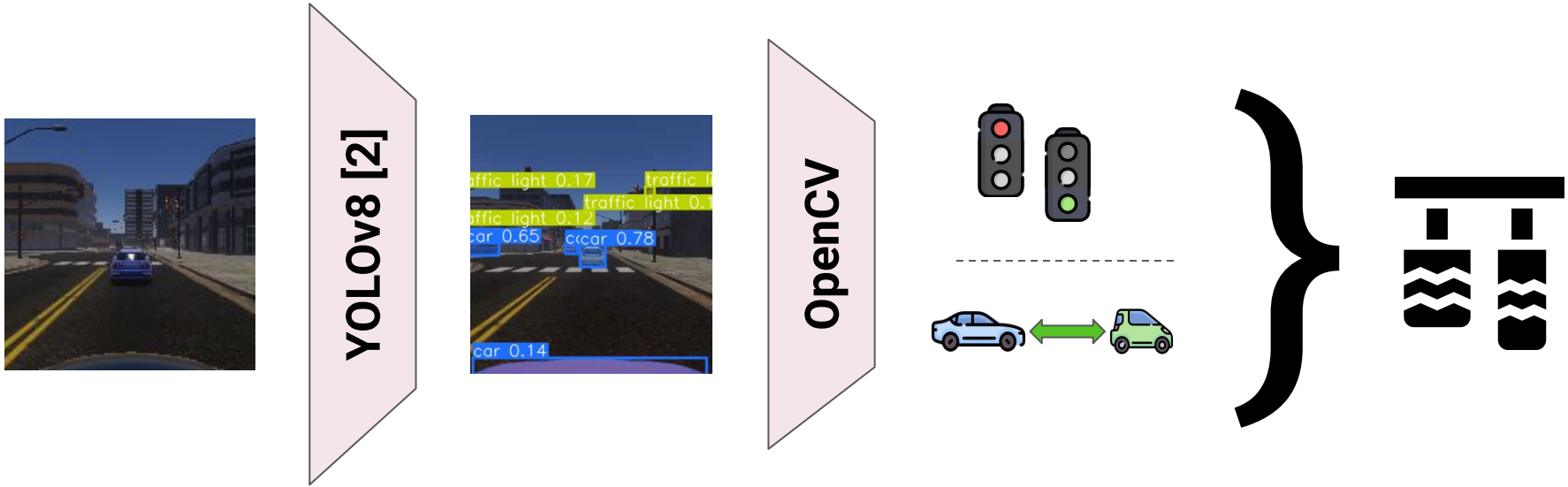
Deployment - Detection

YOLOv8 [2]

Deployment - Detection

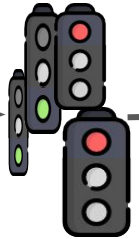


Deployment - Detection

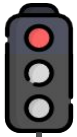


How the Traffic Light Detection works.

Crop Traffic Lights
! Change Thresholds !



HSV Filtering
Find traffic light with most amount of color

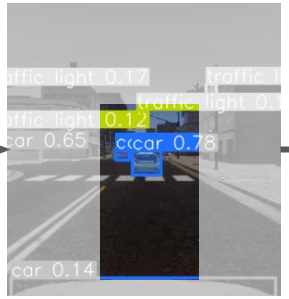


#colored_pixels > threshold?

React to Traffic Light! (e.g. brake / drive)

How the Distance Detection works.

Focus on cars in front

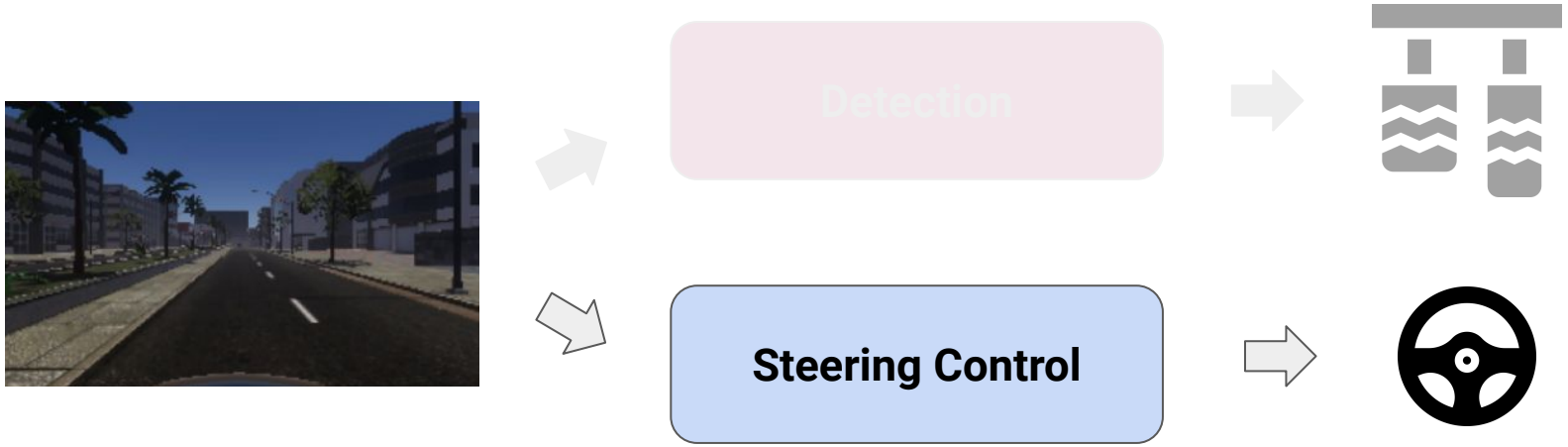


Select Closest Bounding Box
Only Vehicles are Considered

Closest y-Coordinate > threshold?

Brake.

Deployment - Steering Control



Deployment - Steering Control

Deployment - Steering Control

How does the model know where to go?

General Approach

1. Collect Dataset

Images

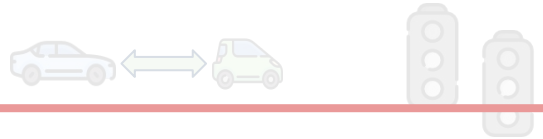
VehicleData

2. Train a model

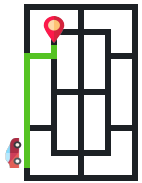
Training

1. Steering Model

2. Detect Surroundings

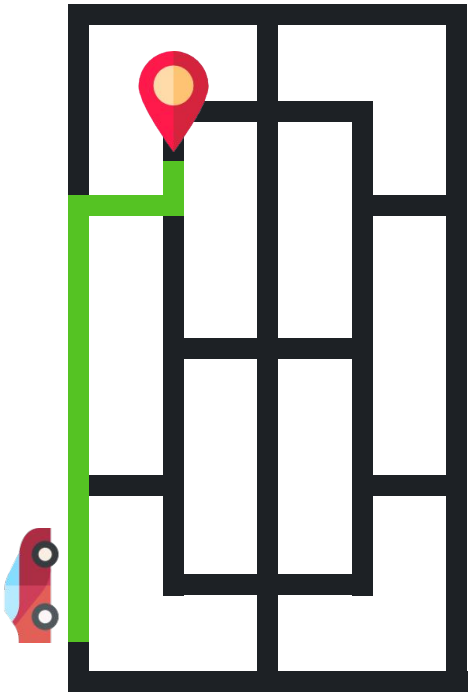


3. Follow Route

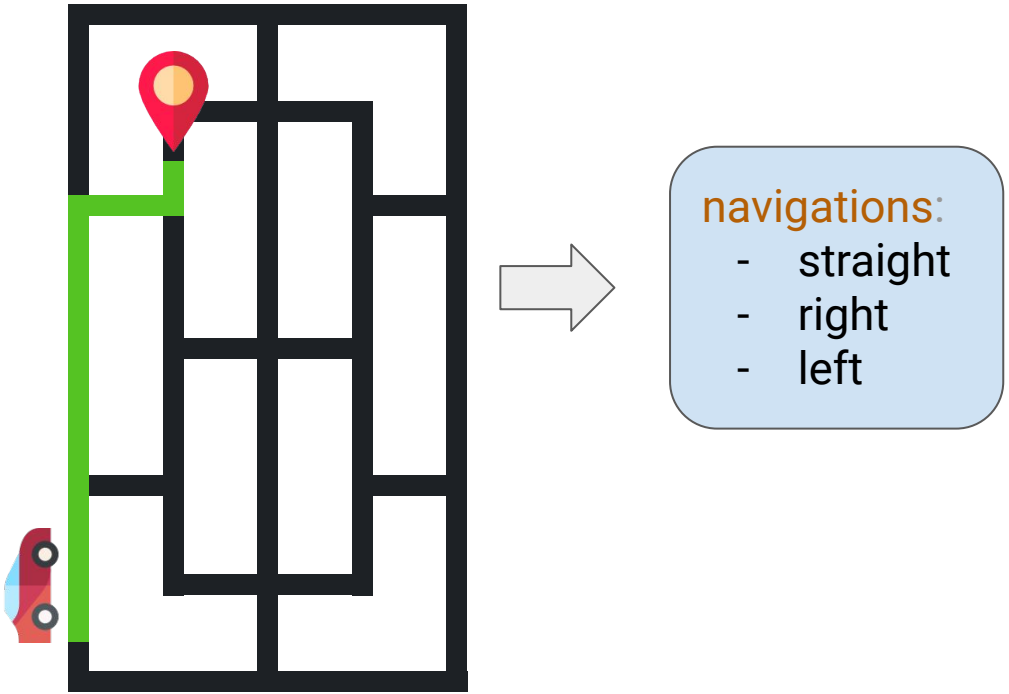


Deployment

Deployment - Navigation

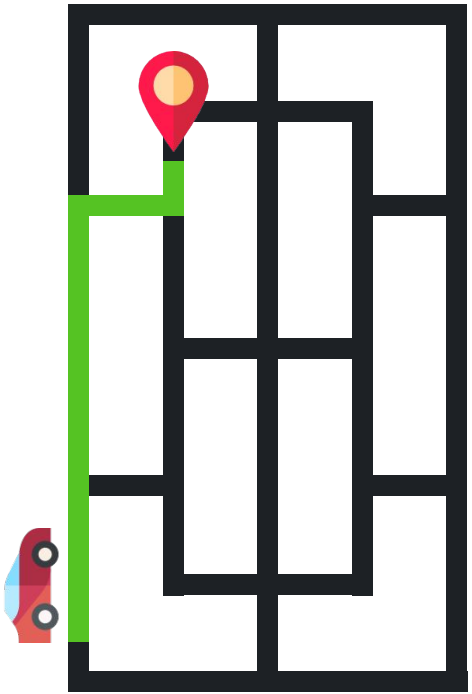


Deployment - Navigation



navigations:
- straight
- right
- left

Deployment - Navigation



navigations:
- straight
- right
- left

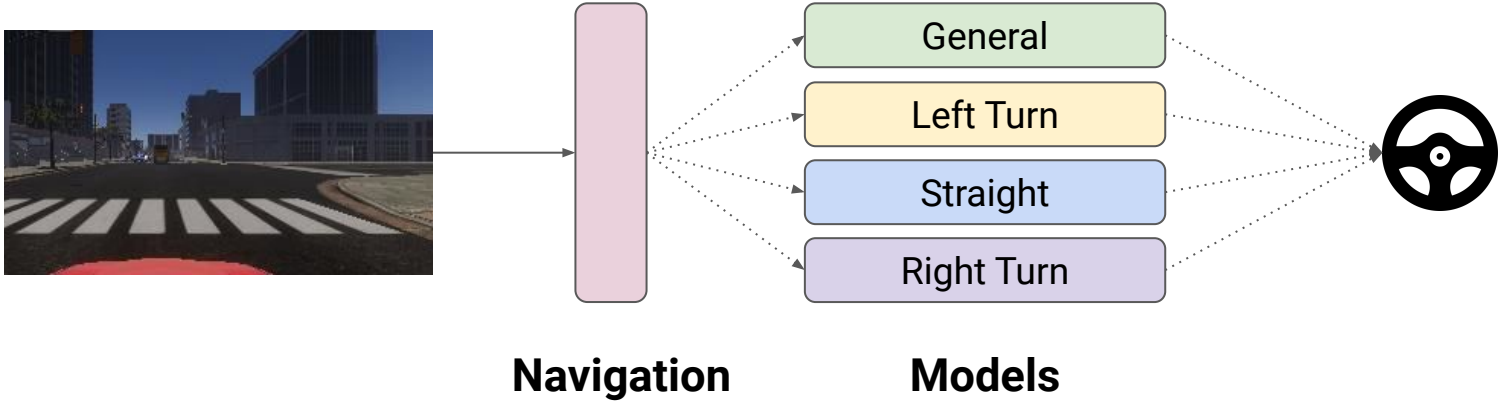
How to follow a route?

Intersection Counter
(Traffic light available?)

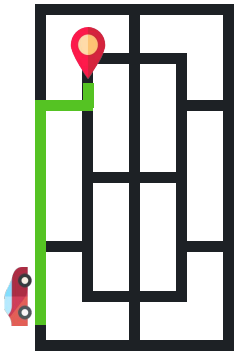
Specific model for each direction

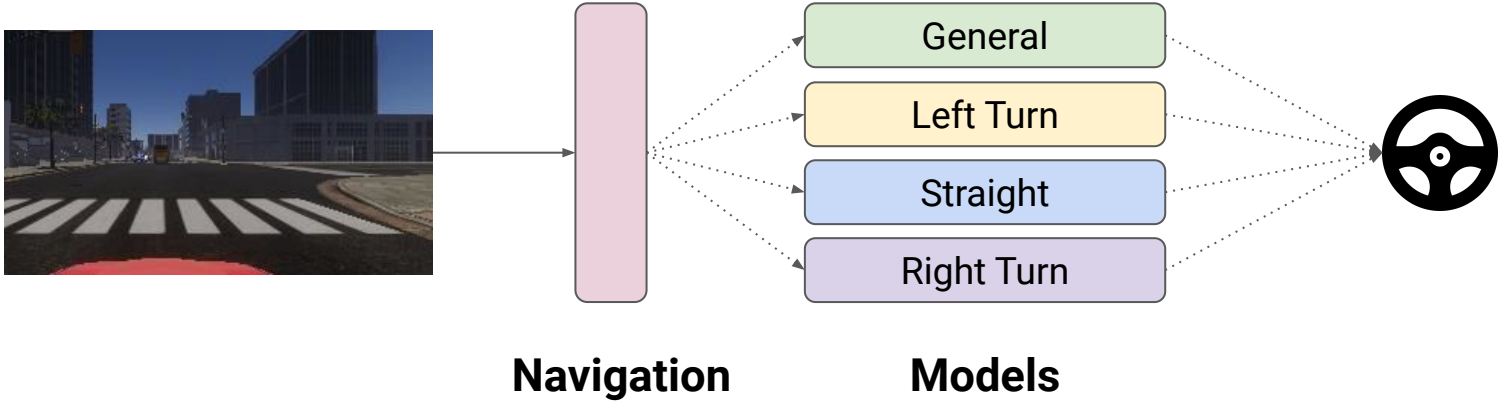
Approach similar to *command-conditional imitation learning* from Codevilla et. al. [1]

Deployment - Navigation

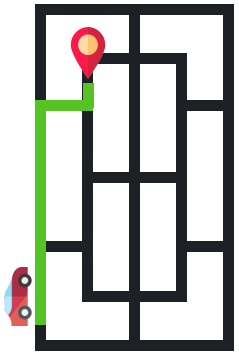


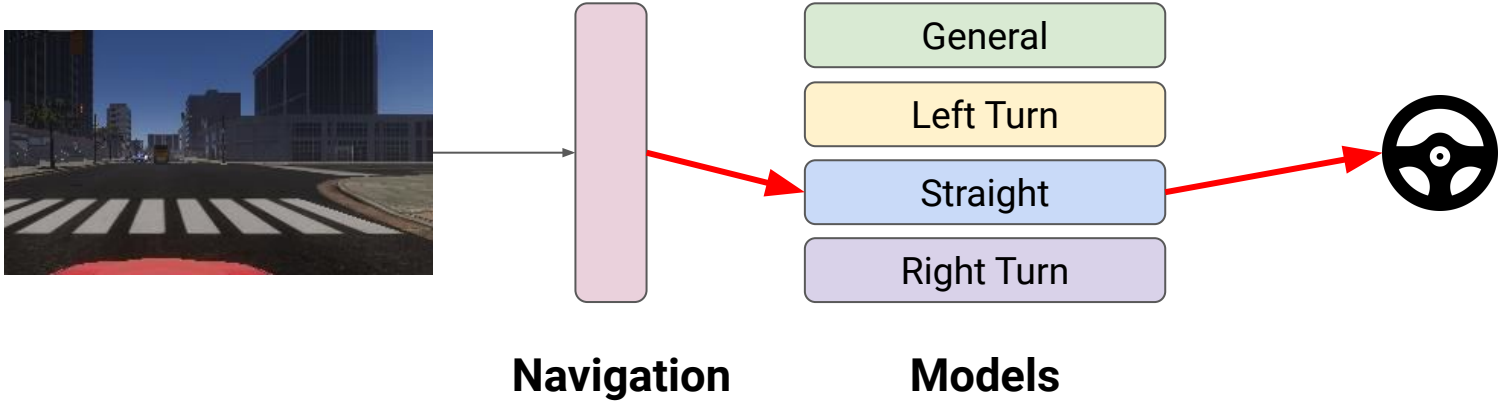
Deployment - Navigation



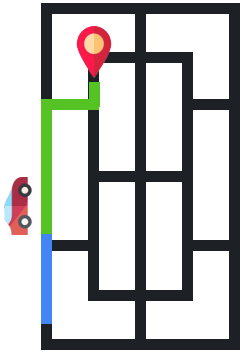


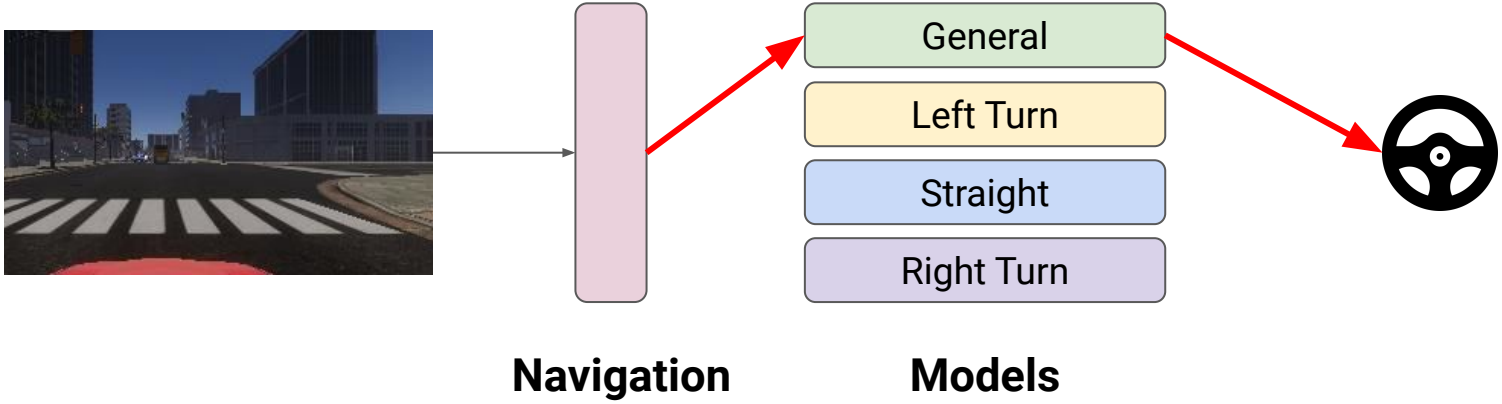
Deployment - Navigation



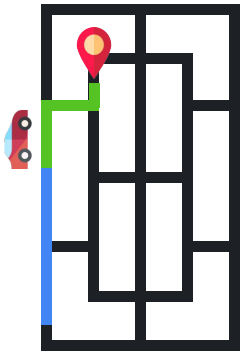


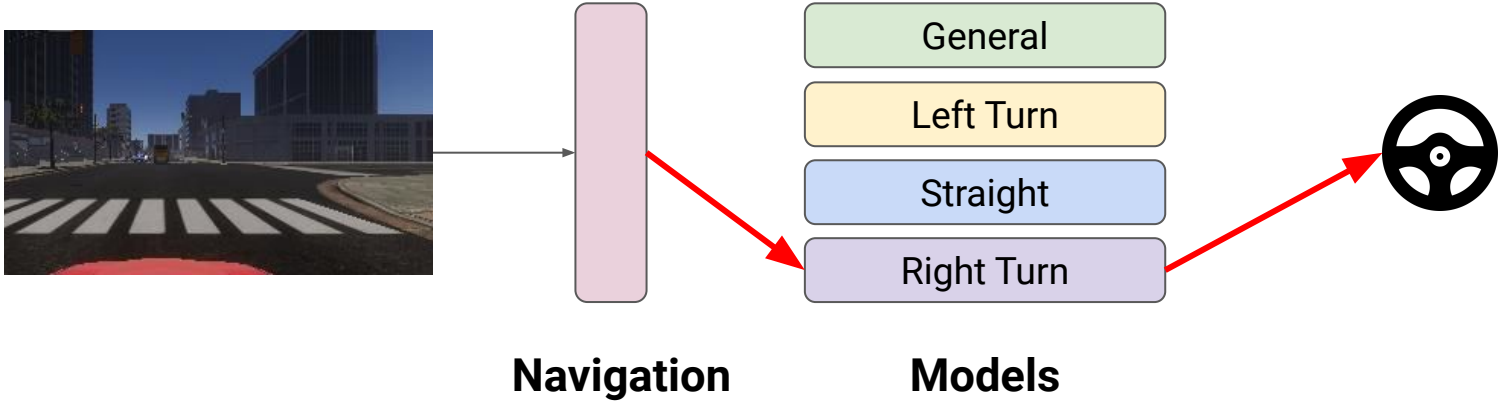
Deployment - Navigation



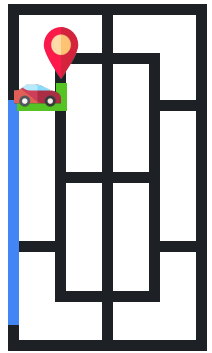


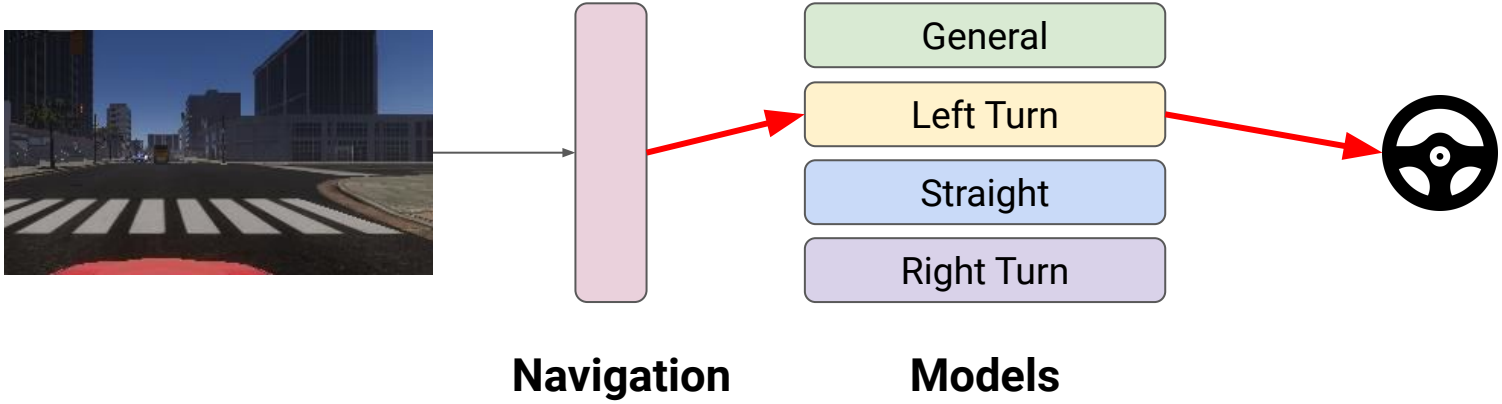
Deployment - Navigation



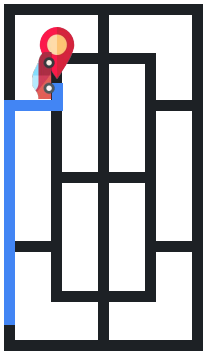


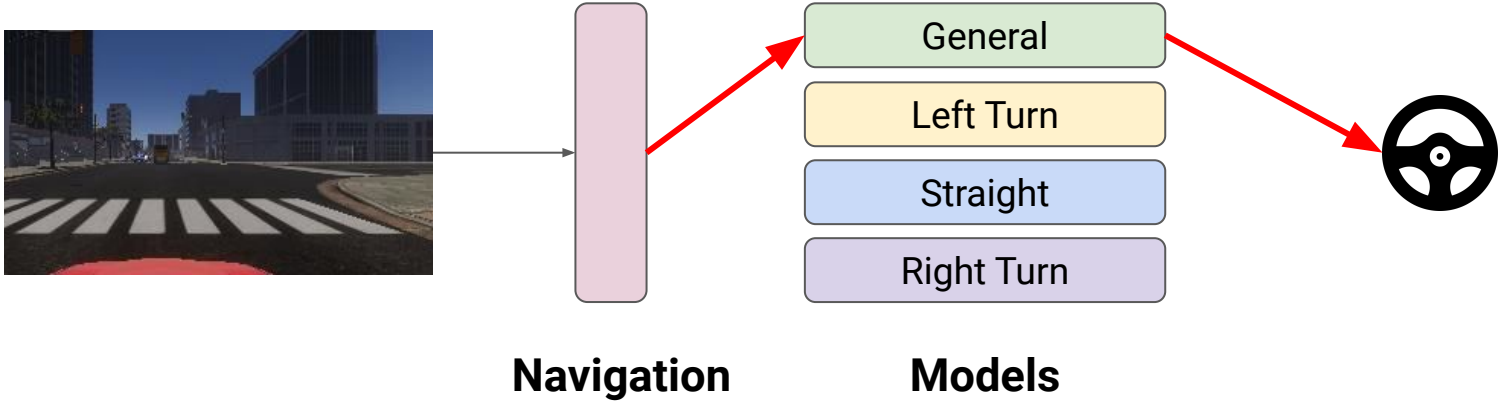
Deployment - Navigation





Deployment - Navigation



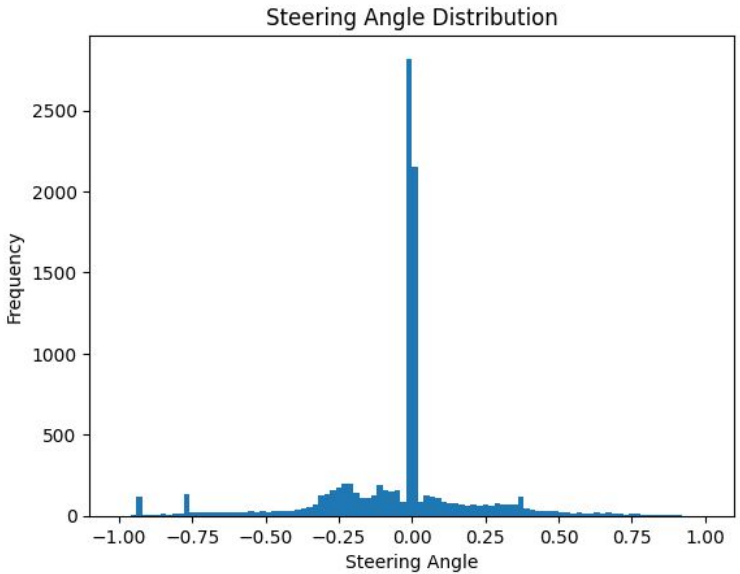


How to train intended Behavior?

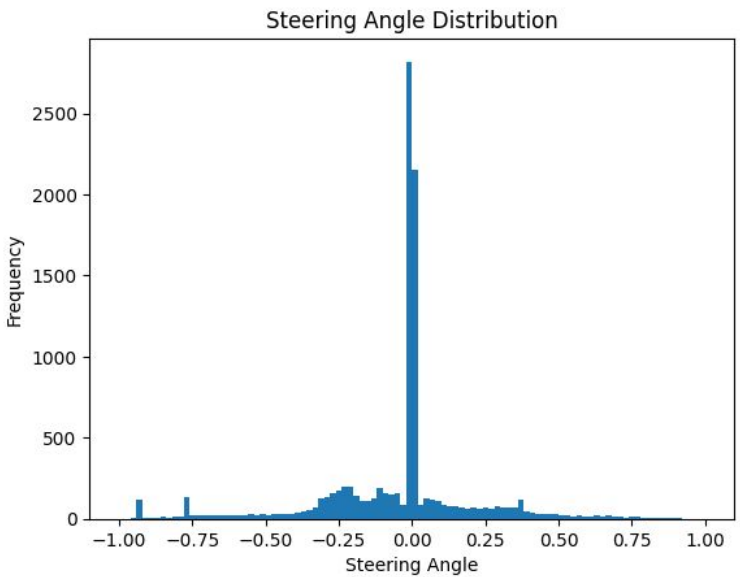
How to train intended Behavior?

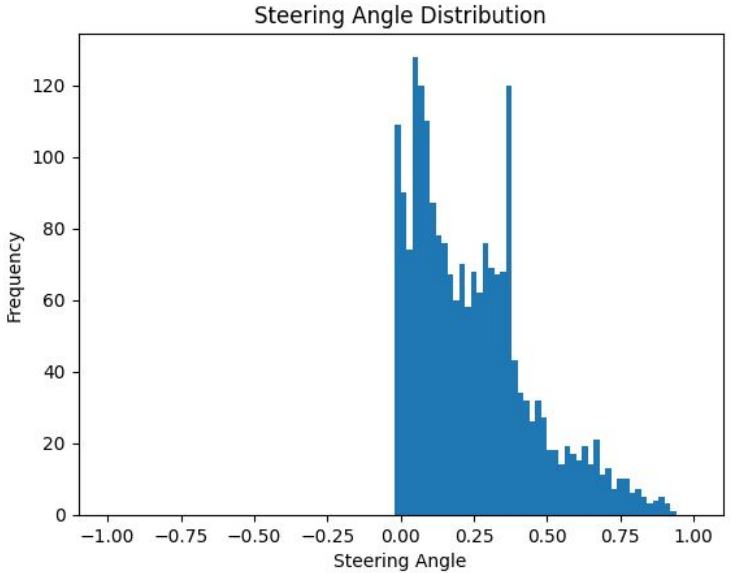
Filter the Training Data.

How to train intended Behavior?



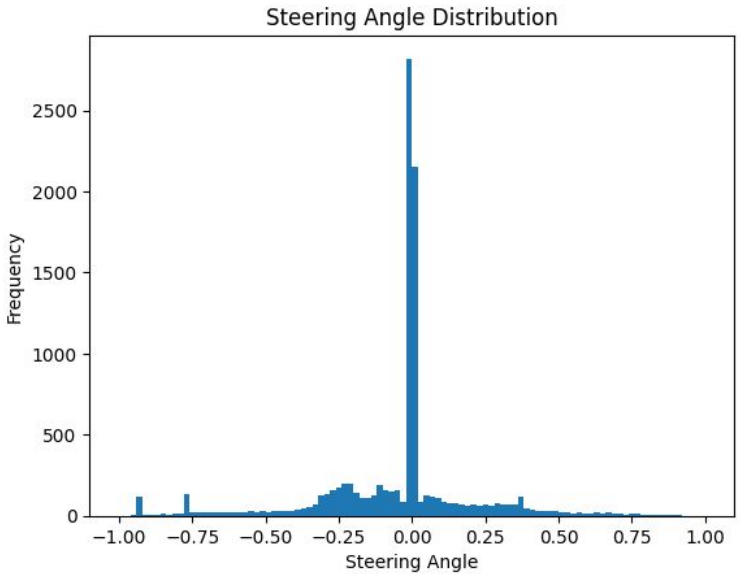
How to train intended Behavior?

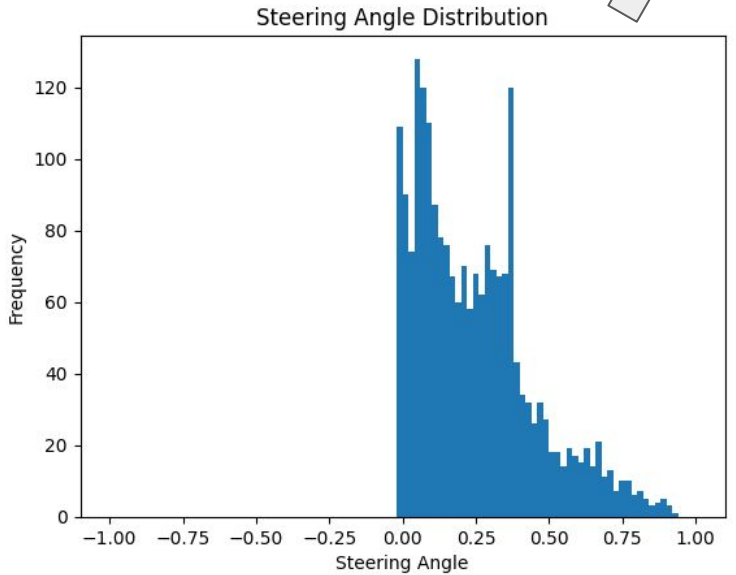




How to train intended Behavior?

Right Turn Model





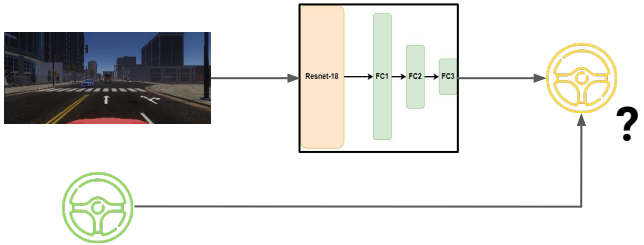
General Approach

1. Collect Dataset

Images

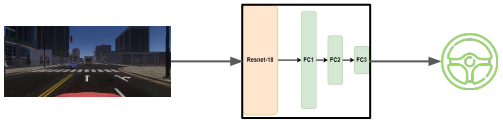
VehicleData

2. Train a model



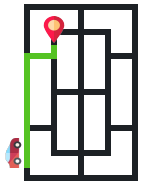
Training

1. Steering Model



2. Detect Surroundings

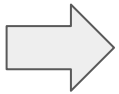
3. Follow Route



Deployment

Evaluation

Navigations	Run 1	Run 2	Run 3
left, left	*PASSED*	*PASSED*	*FAILED*
left, right	*PASSED*	*PASSED*	*PASSED*
straight, left	*PASSED*	*PASSED*	*FAILED*
straight, straight	*FAILED*	*FAILED*	*FAILED*
straight, right	*PASSED*	*FAILED*	*PASSED*
right, left	*PASSED*	*PASSED*	*PASSED*
right, right	*PASSED*	*PASSED*	*PASSED*



35 intersections passed 

6 intersections failed 

1 not reached

Demo

On the unmodified Original Simulator

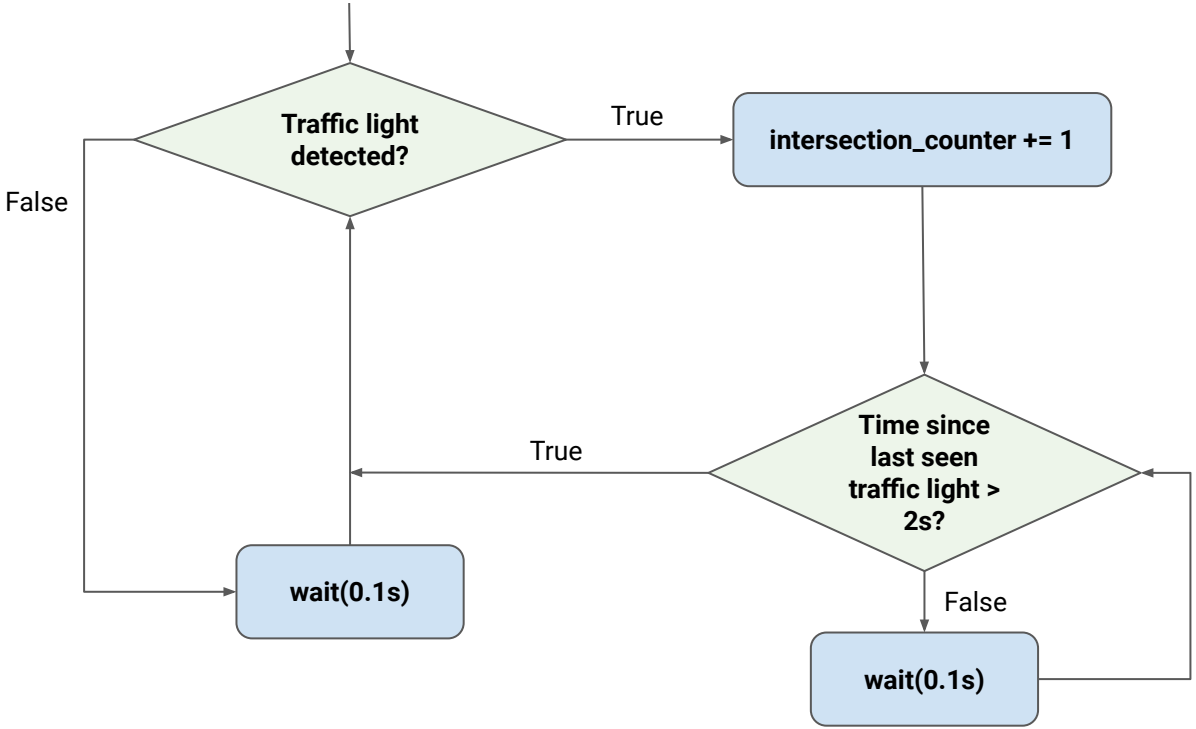
References

[1]: F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, “End-to-end driving via conditional imitation learning,” 2017. [Online]. Available: http://arxiv.org/abs/1710.02410

[2]: Ultralytics, 2023. [Website]. https://ultralytics.com/

Extra Slides

How the Navigation works.



Outlook