AUTONOMOUS DRIVING VIA RL WITH OBJECT DETECTION

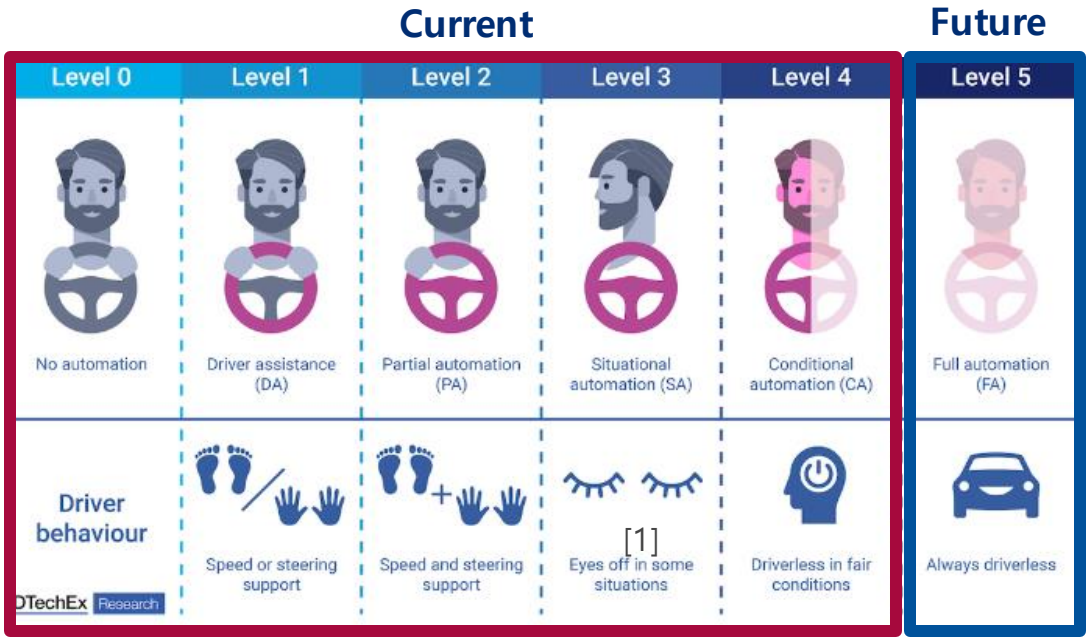
Brandon Fitch
Celina Polanco
Charnay Hatten
Firas Abou Karnib
Ory Wickizer

Agenda

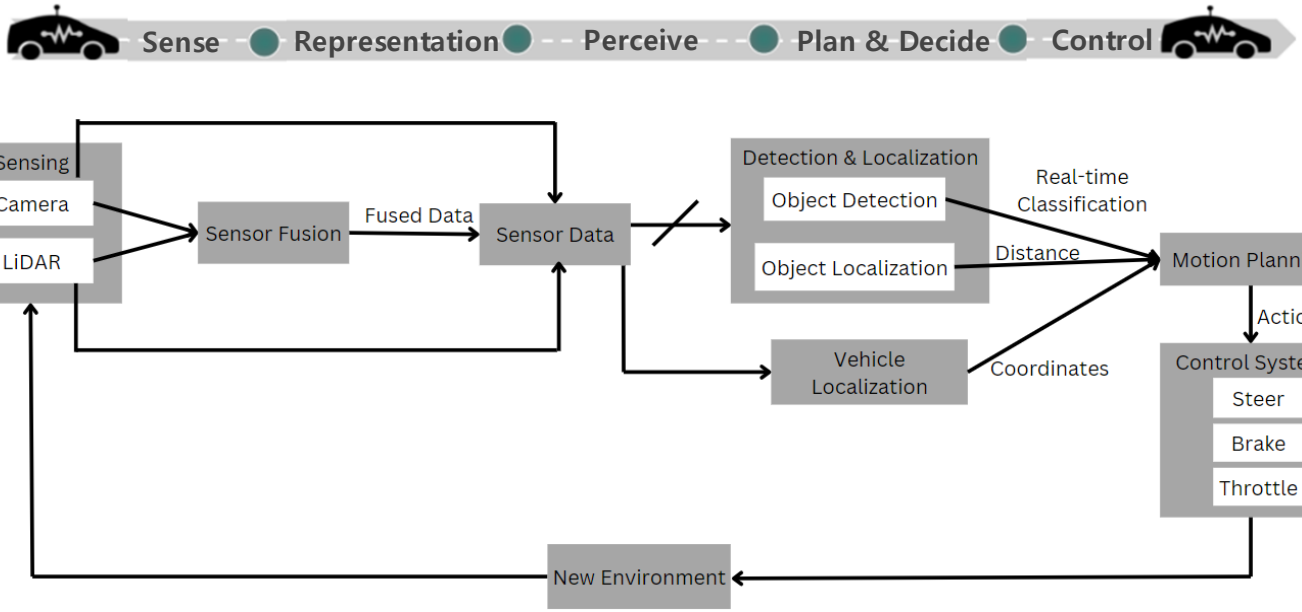
- Project Overview
- Design
  - Requirements
  - Object Detection: YOLO
  - Motion Planning: Reinforcement Learning
- Implementation
  - Simulation Environment: CARLA
  - Final Block Diagram
  - YOLO
  - Reinforcement Learning
- Conclusion
- Next Steps

UNIVERSITY OF DETROIT MERCY

Autonomy Overview



Design-Initial Block Diagram



Design - Requirements

- The system will efficiently detect:

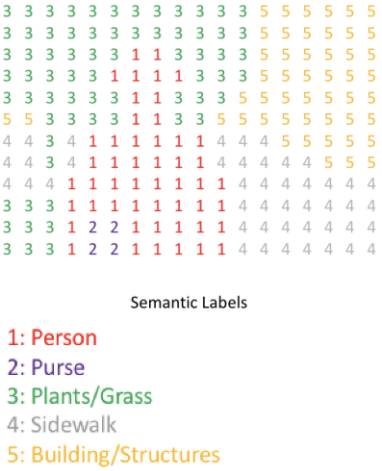
Requirement	Details
Stop at stop signs	Hold for 5 seconds
Stop at red lights	Complete stop
Slow down at yellow lights	Decelerate to a stop
Proceed at green lights	Go
Maneuver around static obstacles	Navigate around pedestrians and vehicles
Penalty for close pedestrian proximity	If closer than 10 ft to a pedestrian
Penalty for close vehicle proximity	If closer than 3 ft to a vehicle
Stop close to stop signs	Must stop within 3 ft

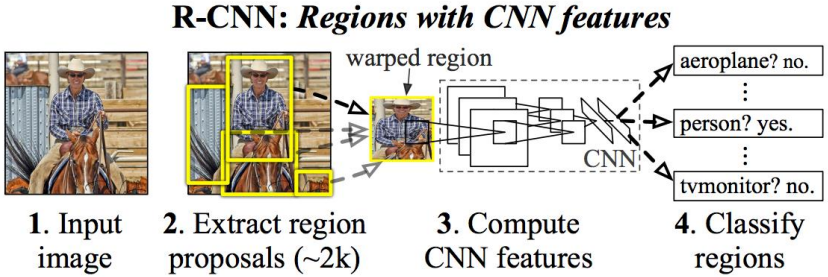
UNIVERSITY OF DETROIT MERCY

SYSTEM DESIGN

Object detection in general

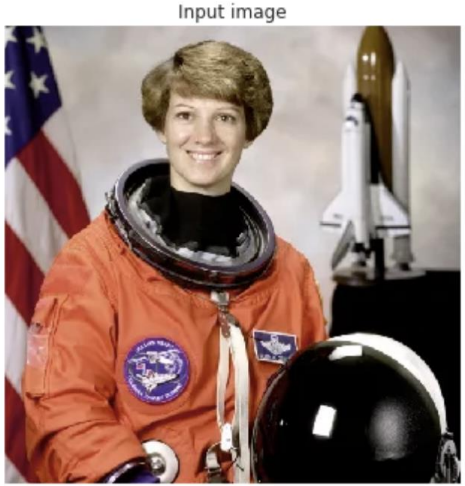
- Current methods
  - Deep Learning Methods
    - Convolution Neural Network (CNN)
    - Region-CNN
    - Semantic Segmentation
  - Lidar and Radar Methods
    - Sensor Fusion

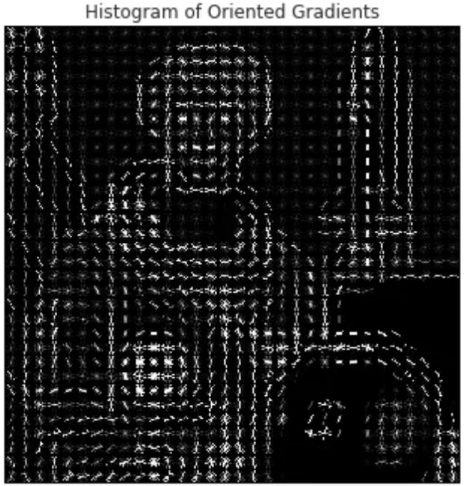




What was available before YOLO?

- Haar Cascade
- Histogram of Oriented Gradient (HOG)

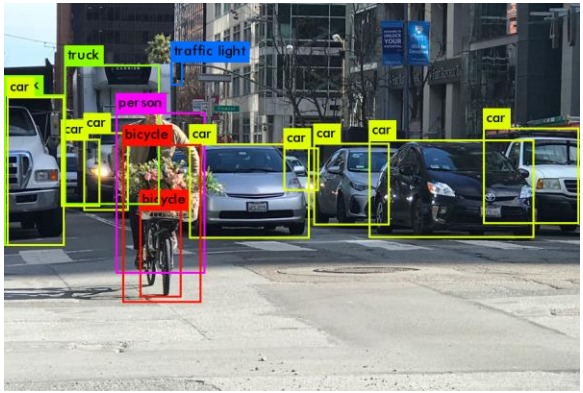




UNIVERSITY OF DETROIT MERCY

Design – Object Detection: YOLO

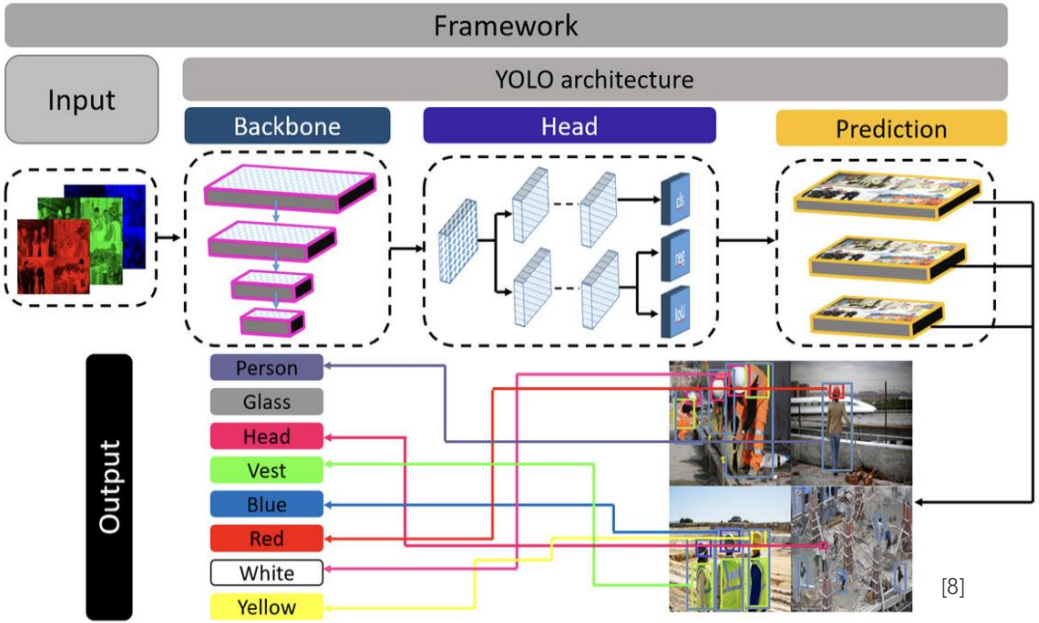
- You Only Look Once
  - Speed
    - Propagates through CNN only once
  - Accuracy
    - Real-time detection
  - End-to-End Learning
    - Better efficiency
    - Better performance



[7]

UNIVERSITY OF DETROIT MERCY

Design – Object Detection: YOLO (cont.)



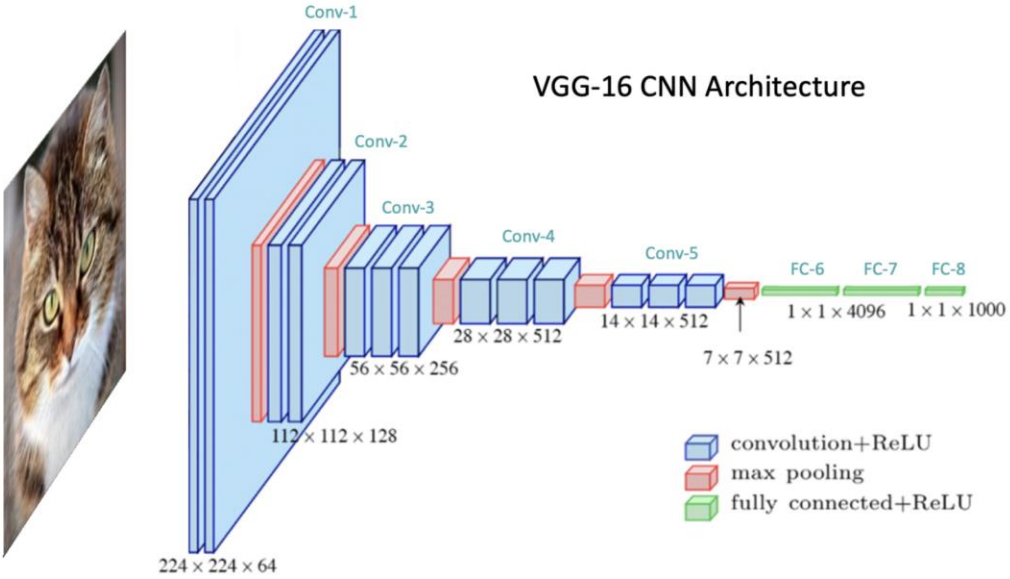
[8]

UNIVERSITY OF DETROIT MERCY

[8] PPE detector: a YOLO-based architecture to detect personal protective equipment (PPE) for construction sites, Md. Ferdous et. Al., 2022

Design – Object Detection: YOLO (cont.)

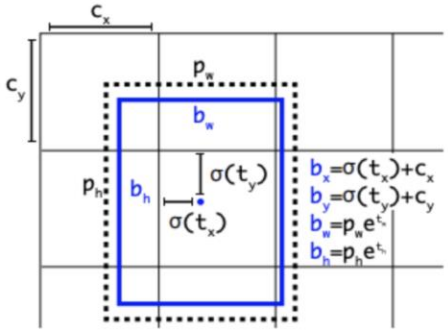
- Convolutional Neural Network (CNN) in YOLO



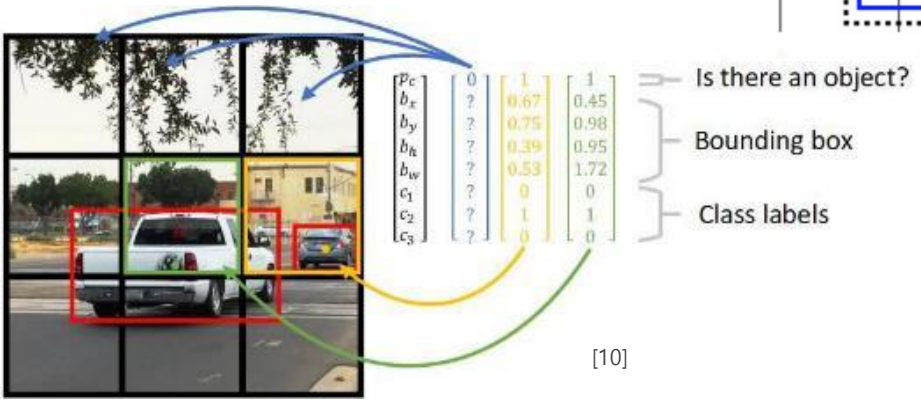
[9]

Design – Object Detection: YOLO (cont.)

- Output data represented as:
  - Bounding boxes
  - Object scores or probabilities
  - Class labels

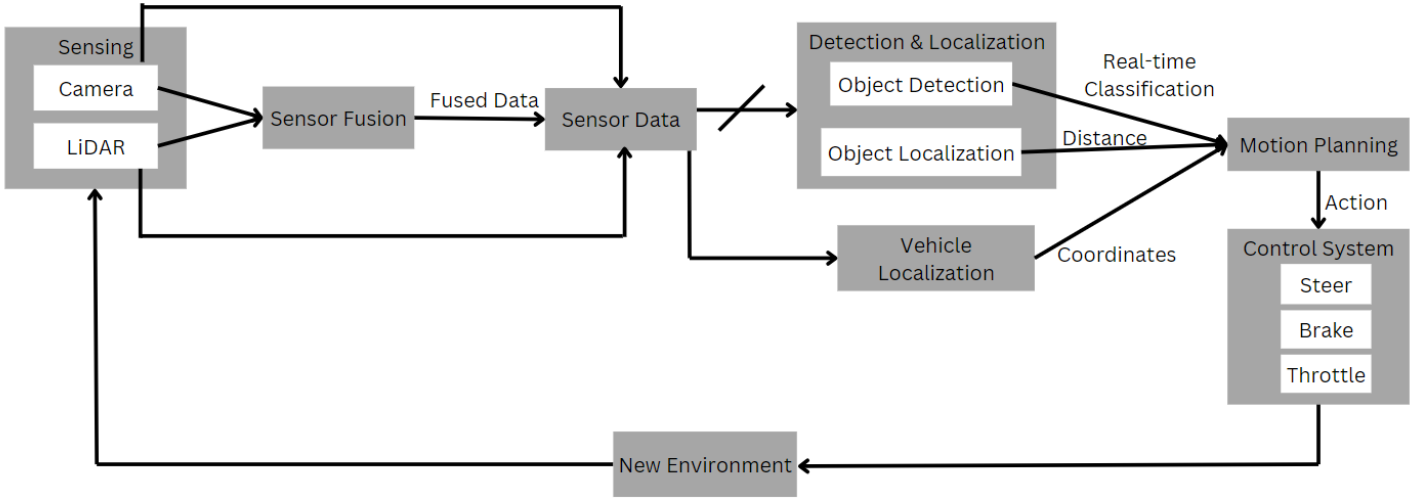


[11]



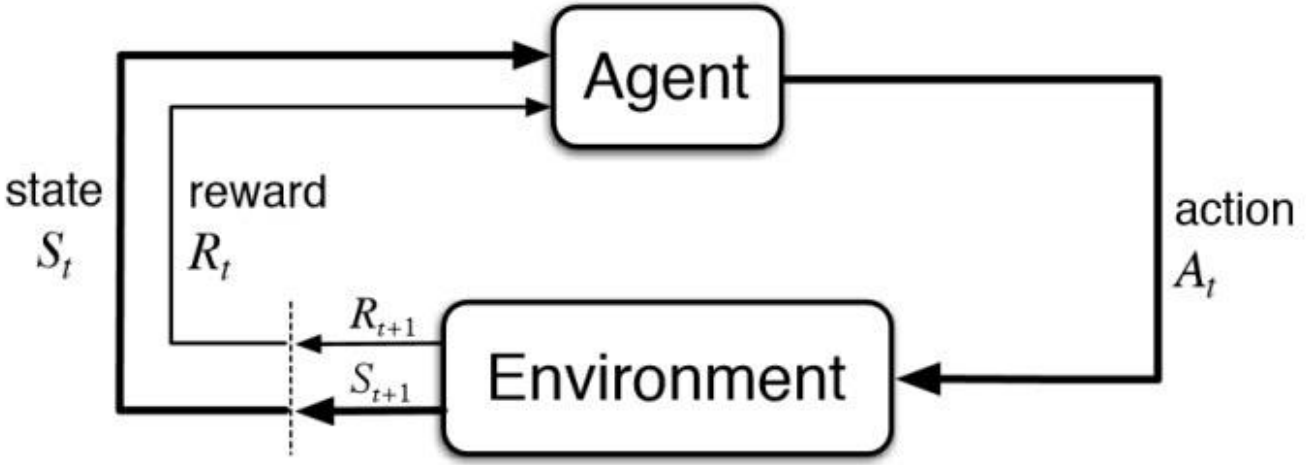
[10]

Design – Motion Planning



Design – Motion Planning: RL

- What is Reinforcement Learning?



Design – Motion Planning: RL

Exploitation

- Taking the highest rewarding action given a state.

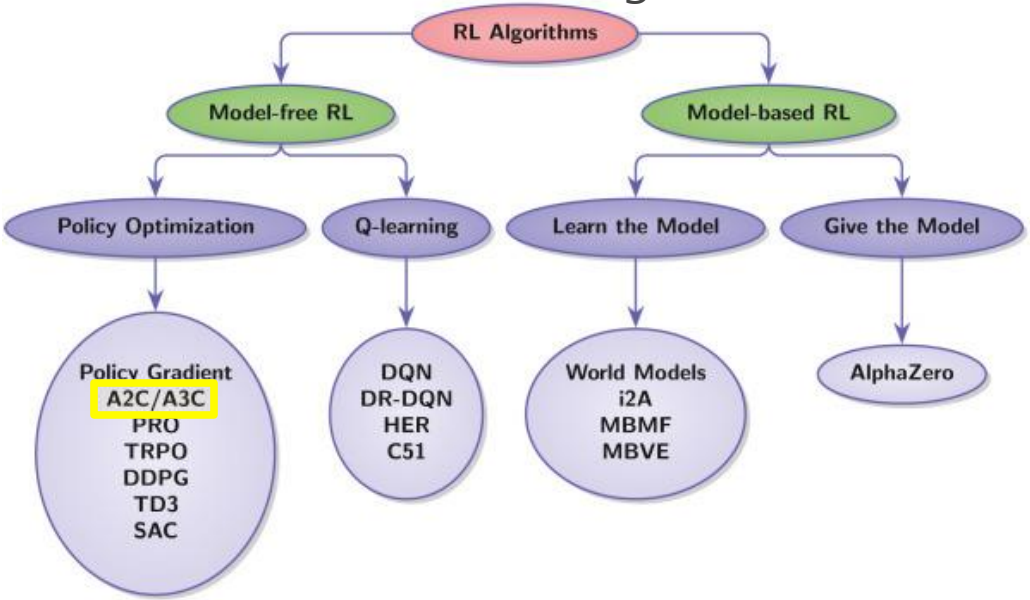
Exploration

- Choosing a random action, no consideration to reward

UNIVERSITY OF DETROIT MERCY

Design – Motion Planning: RL

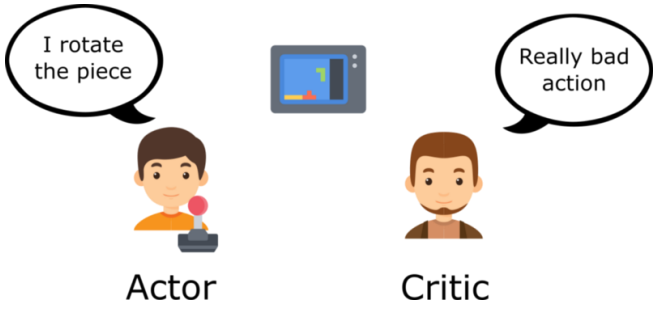
- Types of Reinforcement Learning



UNIVERSITY OF DETROIT MERCY

Design – Motion Planning: RL

- A2C – Advantage Actor Critic



Value Function for current state

$$A(s,a) = r + \gamma V(s') - V(s)$$

Immediate Reward

Discount Factor

Value Function of the next state

UNIVERSITY OF DETROIT MERCY

https://medium.com/free-code-camp/an-intro-to-advantage-actor-critic-methods-lets-play-sonic-the-hedgehog-86d62407171d

Design – Motion Planning: RL

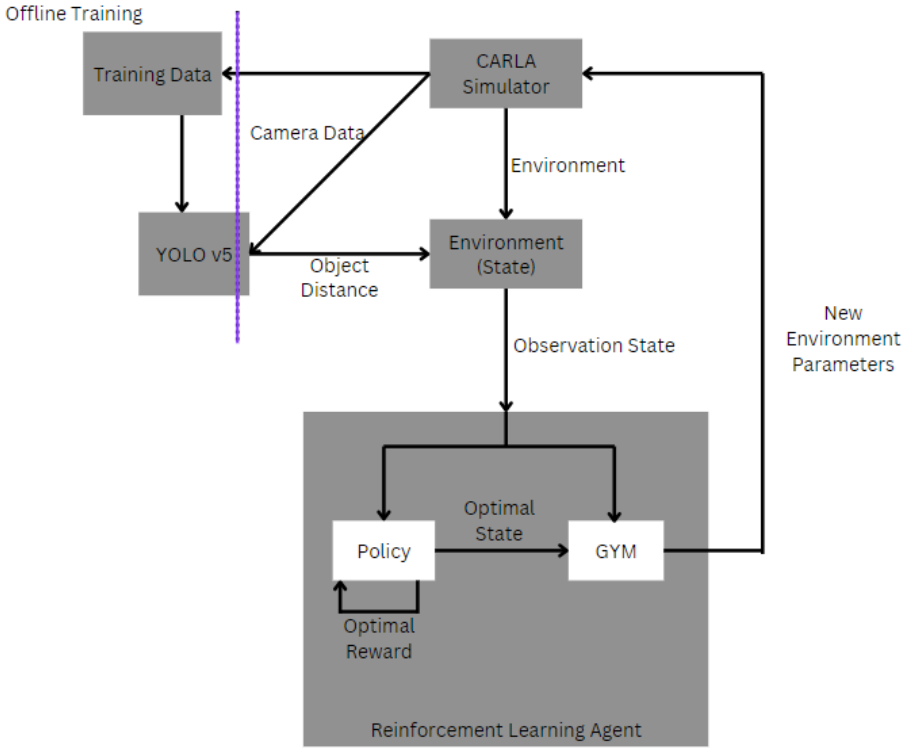
- OpenAI Gymnasium
  - Open-source Python Library that defines a standard API for communication between RL algorithms & environments
  - Provides an interface between the simulator & the agent for online training

UNIVERSITY OF DETROIT MERCY

Design – Simulation Environment: CARLA

UNIVERSITY OF DETROIT MERCY

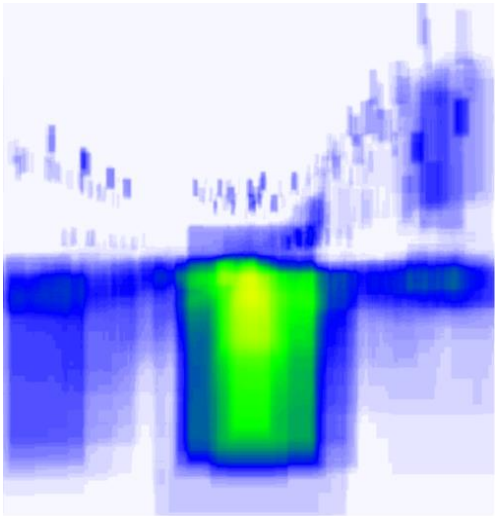
Design – Final Block Diagram



SYSTEM IMPLEMENTATION

Implementation – YOLO

- Carla Detection YOLOv5 code
- Versioning and dependency resolution
- Collecting data and labeling



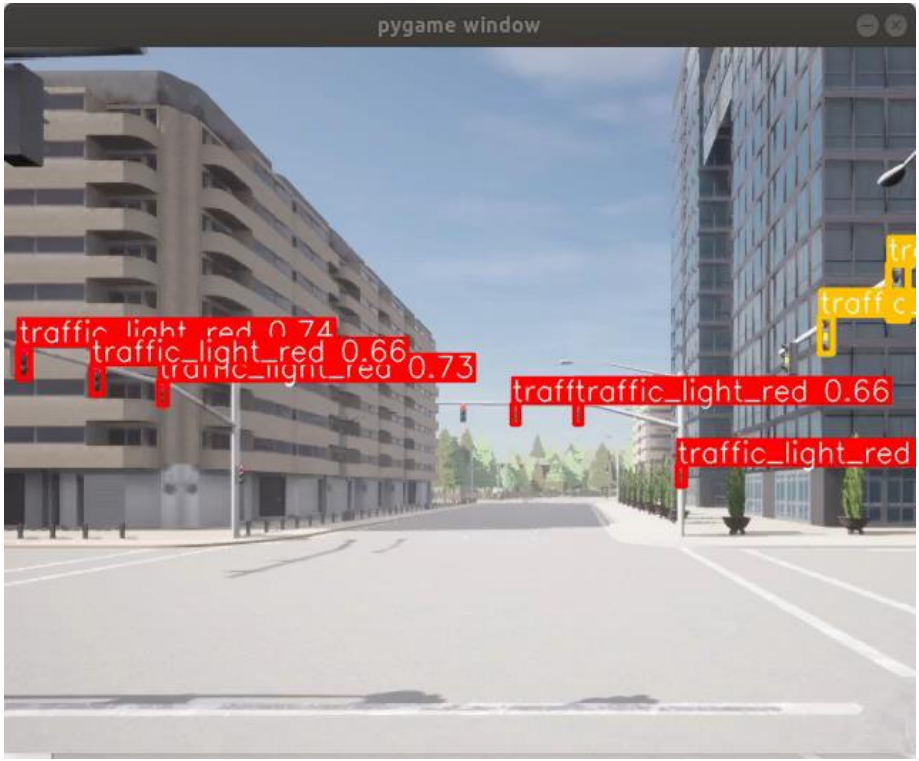
Annotation Heatmap

[14] Alaca, Ismail Ferdi. "Multi Scale Light Weight Road Sign And Crosswalk Detection on CARLA Simulator" Oct. 2023

Implementation – YOLO (Cont.)

UNIVERSITY OF DETROIT MERCY

Implementation – YOLO (Cont.)



UNIVERSITY OF DETROIT MERCY

Implementation – YOLO (Cont.)

YOLOv5:Our Code

Class	Images	Instances	P	R	mAP50	mAP50-95:
all	100	363	0.905	0.844	0.898	0.39
traffic_light_green	100	97	0.908	0.835	0.876	0.397
traffic_light_red	100	94	0.912	0.767	0.881	0.353
traffic_light_yellow	100	172	0.897	0.93	0.938	0.42

YOLOv8:COCO

Format	Status	Size (MB)	metrics/mAP50-95(B)	Inference time (ms/im)
PyTorch	✓	6.2	0.6382	7.14

Implementation – RL

- Working with the packages **Gymnasium** and **Stable Baselines3** as our foundation

- Gymnasium
  - Wrapper for:
    - Making custom environments
    - Allowing integration with SB3
  - One of the research community's go-to packages for RL
    - Gym.make()
    - Gym.reset()
    - Gym.step()

- **Stable Baselines3**
  - Wrapper to train and evaluate agents in Gym environments

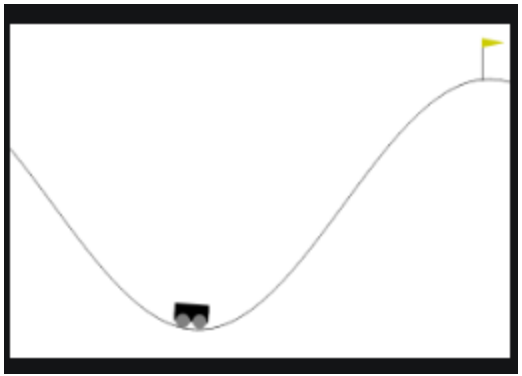
UNIVERSITY OF DETROIT MERCY

Implementation – RL

- Started with simple, pre-made Gym environments to learn the workflow
- Gradually increased to more complex pre-made environments

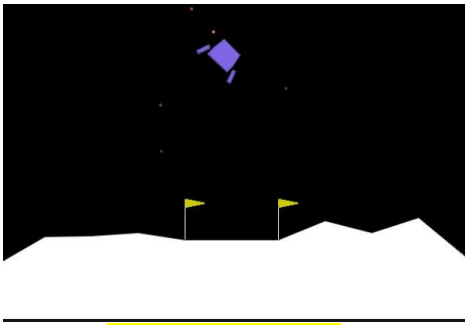
Frozen Lake

Action Space
Discrete | 2D



Mountain Car

Action Space
Continuous | 1D



Lunar Lander

Action Space
Continuous | 2D

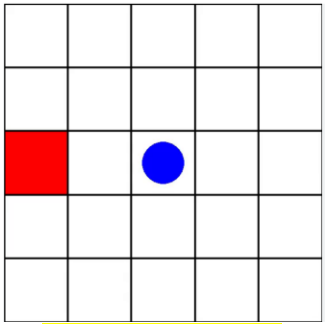
UNIVERSITY OF DETROIT MERCY

Implementation – RL (Cont.)

- Increased custom environment complexity, until we arrived at gym-carla

```
Step 9
Action:  [0]
obs= [[9.]] reward= [1.] done= [ True]
.........x.
Goal reached! reward= [1.]
```

Gridworld 1D



Gridworld 2D

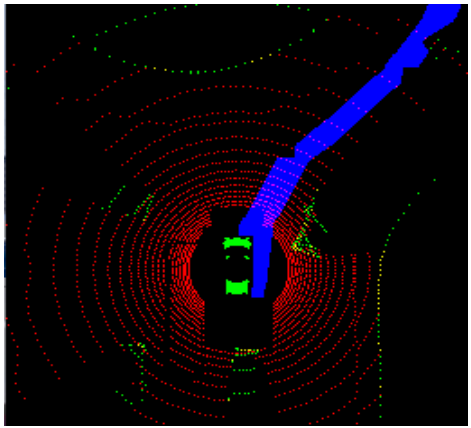
gym-carla
[5]

UNIVERSITY OF DETROIT MERCY

Implementation – RL (Cont.)

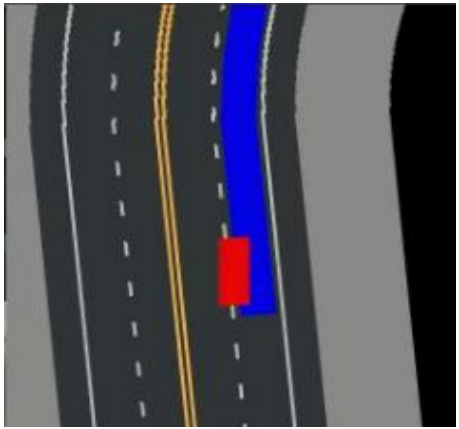
- **Observation space**

RGB Camera Image



Lidar Point Cloud

Vehicle State
(Position, Heading, Velocity)

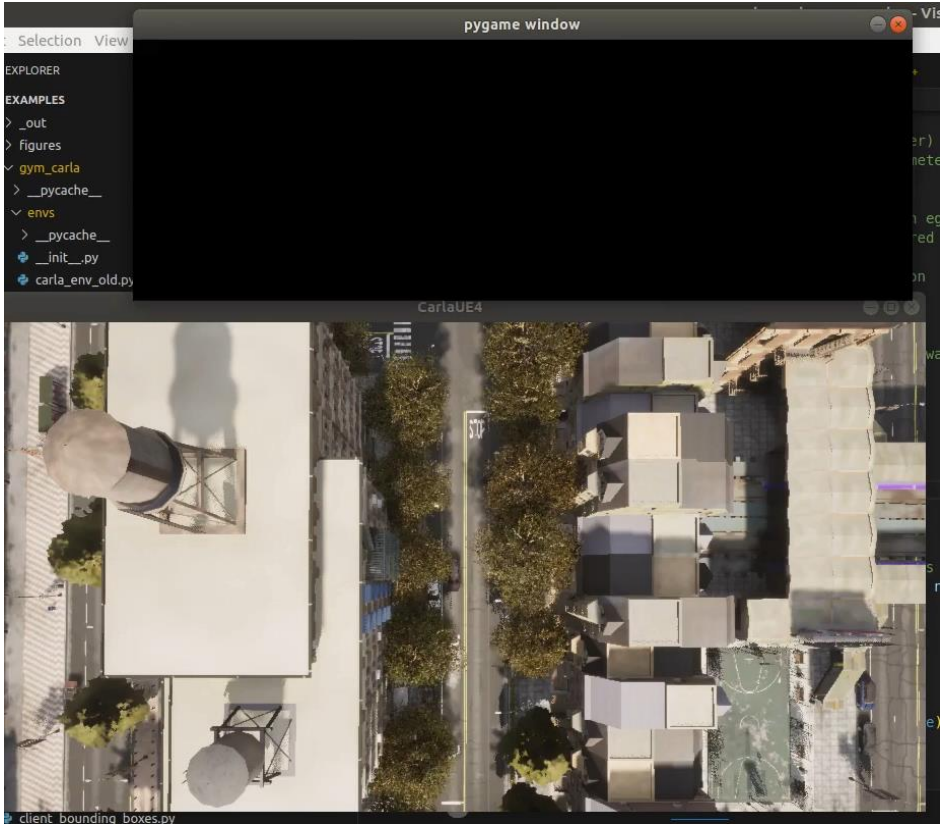


Bird's Eye View Rendering

- **Action space**
  - Throttle Amount
  - Steering Angle

UNIVERSITY OF DETROIT MERCY

Implementation – RL (Cont.)



Implementation – RL (Cont.)

- Example Step of an action taken

```
time_step = 2
total_step = 10010

reward: -0.5500000260456676
collision reward: 0
longitudinal speed reward: 9.717120006561141e-09
too fast reward: 0
out of lane reward: 0
steering reward: -0.4500000357627876
lateral acceleration reward: -5.6653454984352416e-18
action taken: [3.         0.22151384]
```

- Reward Function

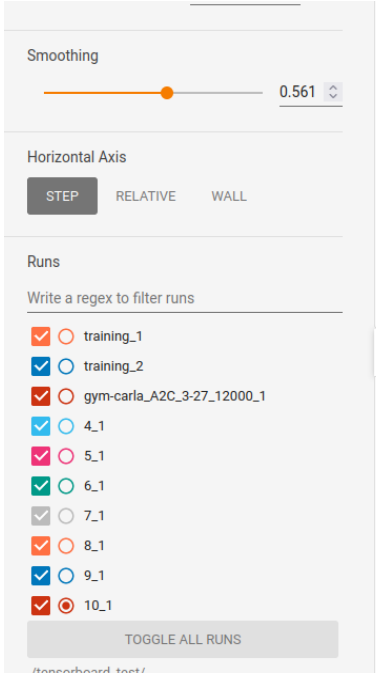
R = - a * has collided with object?
+ b * in the same direction as waypoint?
- c * is going too fast?
- d * is out of lane?
- e * is steering sharply?
- f * is accelerating laterally?
- 1 existential dread

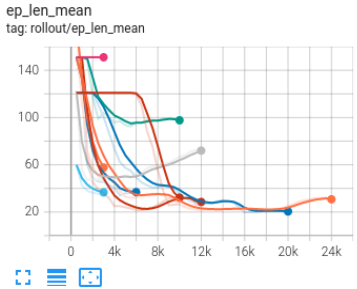
UNIVERSITY OF DETROIT MERCY

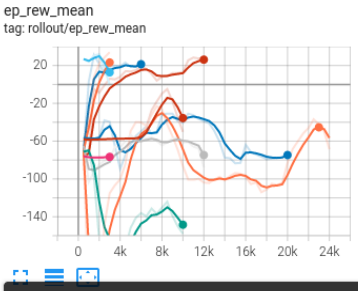
Implementation – RL (Cont.)

- A look at some earlier training runs

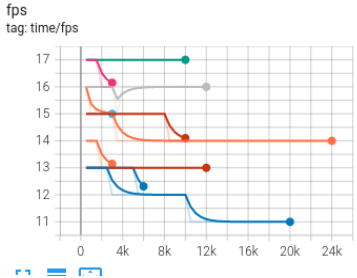
*different length training runs included as an example of step size as a hyperparameter







time



Name	Smoothed	Value	Step	Time	Relative
10_1	-35.6	-50.14	10k	Tue Apr 2, 10:45:02	10m 44s
4_1	12.78	0.1349	3k	Wed Mar 27, 14:15:14	2m 47s
5_1	-76.68	-76.17	3k	Wed Mar 27, 14:21:03	2m 28s
6_1	-148.4	-157.9	10k	Wed Mar 27, 14:32:52	9m 2s
7_1	-74.73	-87.43	12k	Wed Mar 27, 14:56:43	11m 52s
8_1	-45.35	-42.45	23k	Wed Mar 27, 15:37:55	26m 23s
9_1	-74.7	-71.36	20k	Thu Mar 28, 12:05:26	28m 32s
gym-carla_A2C_3-27_12000_1	26.21	28.18	12k	Wed Mar 27, 13:59:46	14m 37s
training_1	23.22	34.01	3k	Wed Mar 27, 13:15:16	3m 1s
training_2	21.36	24.36	6k	Wed Mar 27, 13:31:49	7m 5s

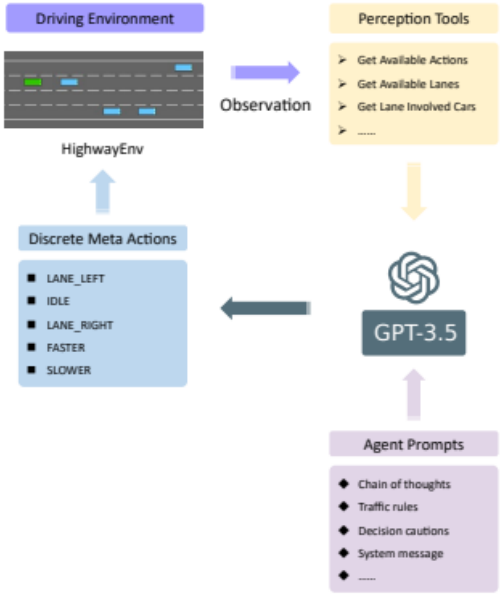
Implementation – RL (Cont.)

- Hyperparameter tuning

Lighter color = Lower number
Darker color = Higher number

	A	B	E	F	G	H	I	J	K	L	M	N	O	P
	tensorboard name	what it does												
	model name		gym-carla_A2	4	5	6	7	8	9	10	11	12	13	14
	hyperparameter													
	training steps		12000	3000	3000	10000	12000	24000	20000	10000	10000	7000	9000	20000
	max episode time		150	150	150	150	150	200	120	120	120	120	120	120
	out_lane_thres		2	4	4	4	4	4	2	2	2	2	2	2
	desired_speed		8	8	4	4	4	4	4	8	8	8	8	8
	dt			0.1	0.01	0.05	0.1	0.1	0.4	0.1	0.1	0.1	0.1	0.1
	max_waypt		12	12	12	12	12	12	12	12	12	12	12	12
	discrete_acc		-3,3	-3,3	-3,3	-3,3	-3,3	-3,3	-3,3	-3,3	-3,3	-3,3	-3,3	-3,3
	r_collision coeff	if it hits something	200	200	200	200	200	200	200	200	200	200	200	200
	speed_lon	if vehicle orientation is match	1	1	1	1	1	1	1	1	1	1	1	1
	r_fast	going too fast	10	10	10	10	10	10	10	10	10	10	10	10
	r_out	out of lane	1	1	1	1	1	1	1	5	5	5	5	5
	r_steer	big steering angle	5	5	5	5	5	5	5	5	5	5	5	5
	r_lat	lateral acceleration	0.2	0.2	0.2	0.2	0.3	0.3	0.2	0.2	0.2	0.2	0.2	0.2
	dread		0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	learning_rate		0.0007	0.0007	0.0007	0.0007	0.0007	0.0007	0.0007	0.0007	0.01	0.001	0.001	0.003
	ent_coef		0	0	0	0	0	0	0	0	0	0.1	0.05	0.05
	gamma		0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.999
	n_steps		5	5	5	5	5	5	5	5	5	5	5	5
	notes		just goes righ	mostly just goes right			best as of	mostly jus	just turns r	pretty good			good	just turning
	rating						6		3	6			6	3

Next Steps



GPT-3.5

Thought 1: Since I am encounter such situ, the first thing to do is to check my available actions.
Act 1: Get_available_actions('ego')
Obs 1: [...] The vehicle can do IDLE, FASTER, SLOWER, LANE CHANGE RIGHT actions. [...]

Thought 2: Now I need to check the status of 'ego' car.
Act 2: Get_available_lanes('ego')
Obs 2: [...] The 'ego' can drive on lane_3 and lane_2.Lane_3 is the curent lane and lane_2 is left lane.[...]

Thought 3: Now I need to check for acceleration action.
Act 3: Get_lane_cars('lane_3')
Obs 3: veh_6 is driving in front of 'ego' with 10.2m/s. [...]

Thought 4: I have to check if acceleration is safe with veh_6.
Act 4: Is_action_safe('acc','veh_6')
Obs 4: Acceleration may be conflict with veh_6. [...]

Thought 5: Now I need to check for keep speed action.
Act 5: Get_lane_cars('lane_2')
Obs 5: veh_4 is driving in front of 'ego' with 13.89m/s on lane_2. [...]

.....

Thought 8: [...] I suggest it's safe to keep lane with current speed
Act 8: Finish.

Final Answer:
Decision: IDLE
Explanation: Since the last step is ... I suggest...

[6]

- Using Multimodal LLMs in self-driving
  - Zero or few-shot
    - Don't need hundreds, thousands of runs
  - Model can reason and explain
    - In RL, can be difficult to understand why the model has chosen its actions

UNIVERSITY OF DETROIT MERCY

Conclusion

Research overview

Key takeaways

Future implementations

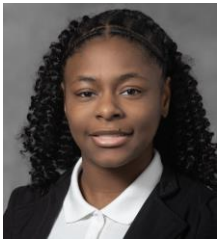
UNIVERSITY OF DETROIT MERCY

Thank you!

Brandon Fitch
Fitchbd1@udmercy.edu

Celina Polanco
Polancci@udmercy.edu

Charnay Hatten
Hattench@udmercy.edu



Firas Abou Karnib
Aboukafi@udmercy.edu

Ory Wickizer
Wickizoj@udmercy.edu

UNIVERSITY OF DETROIT MERCY

Content Links

- Research Paper
  - https://udmercy0-my.sharepoint.com/:w:/g/personal/polancci_udmercy_edu/EUgCvRfJGVZFsHgdcfWpevIBWne-qtQboY9VTccJ11pmZQ?e=W0nLlN
- Github
  - https://github.com/legendairytri/YOLO-RL-CARLA

References

- [1] Connected World. 'The Autonomous Road Ahead.' Connected World, 5 Nov. 2023,
- [2] Deep Reinforcement Learning for Autonomous Driving: A Survey, B Ravi Kiran et. Al., 2021
- [3] CARLA: An Open Urban Driving Simulator, Alexey Dosovitskiy et. Al., 2017
- [4] OpenAI Gym, Greg Brockman, et. Al., 2016
- [5] Interpretable End-to-end Urban Autonomous Driving with Latent Deep Reinforcement Learning, Jianyu Chen et. Al., 2020

UNIVERSITY OF DETROIT MERCY

References (Cont.)

- [6] Drive Like a Human: Rethinking Autonomous Driving with Large Language Models, Daocheng Fu et. Al., 2023
- [7] 10 Things You Need To Know About Ultralytics YOLOv8, Abirami Vina et. Al., 2023
- [8] YOLO(You Only Look Once), Kevin Velasco, 2019
- [9] PPE detector: a YOLO-based architecture to detect personal protective equipment (PPE) for construction sites, Md. Ferdous et. Al., 2022
- [10] Undertanding Colvolutional Neural Network (CNN): A Complete Guide, Sumith Kulal et. Al., 2023

UNIVERSITY OF DETROIT MERCY

References (Cont.)

- [11] YOLO object detection: how does the algorithm predict bounding boxes larger than a grid cell?, Krishnab et. Al., 2018
- [12] YOLO 2 Explained, Zixuan Zhang, 2020
- [13] Lozé, Sébastien. "CARLA Democratizes Autonomous Vehicle R&D with Free Open-Source Simulator." Unreal Engine, 4 Oct. 2019,
- [14] Alaca, Ismail Ferdi. "Multi Scale Light Weight Road Sign And Crosswalk Detection on CARLA Simulator" Oct. 2023

UNIVERSITY OF DETROIT MERCY