Universidad Rey Juan Carlos

RoboticsLabURJC
Programming Robot Intelligence

End-to-end Vision-based Autonomous Driving using Deep Learning

Author
Sergio Paniego Blanco
Supervisor
José María Cañas Plaza
Doctoral Program in Information and Communications Technologies
International Doctoral School
2024
@sergiopaniego
sergio.paniego@urjc.es
https://sergiopaniego.github.io/

Outline

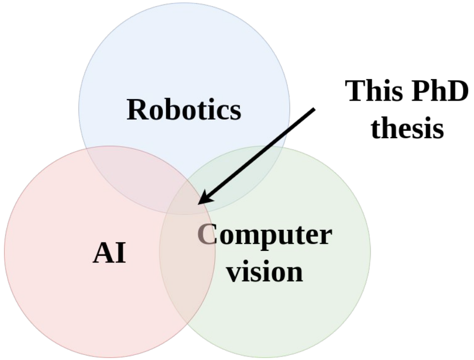
- Introduction
- State-of-the-art
- TrafficSensor
- DetectionMetrics
- Behavior Metrics
- Adding memory
- Optimizing DL models
- Driving in traffic
- Conclusions and future research

Outline

- **Introduction**
- State-of-the-art
- TrafficSensor
- DetectionMetrics
- Behavior Metrics
- Adding memory
- Optimizing DL models
- Driving in traffic
- Conclusions and future research

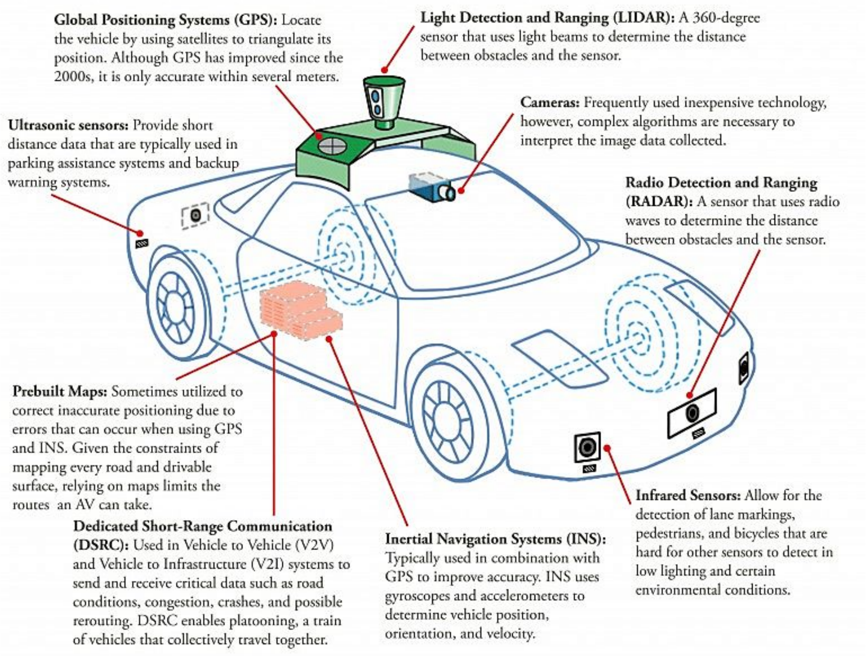
Introduction

- **Autonomous driving (AD):** vehicles that drive safely without human intervention.
- **Robotics + Artificial Intelligence (AI) + Computer Vision (CV).**
- **Applied to AD.**
- Development comes from different sides, specially AI.
- **Benefits and motivation:**
  - Improved traffic safety (94% of accidents caused by human driver error).
  - Optimized mobility.
  - Reduce driver's stress.
  - Reduce operative costs.
  - Reduce pollution.
  - …
- Examples of AD companies suggest real advancement.



Introduction to AD

- Mobile robot in terms of hardware and software.
- Sensors like GPS, IMUs, LIDARS, **cameras**…
- Actuators like throttle, brake, steering…
- CPUs and GPUs.



Levels of autonomy

LEVEL 0	LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5
No automation	One automatic task at a time	Two automatic tasks at a time	The vehicle controls most driving tasks	The vehicle controls all driving tasks under certain conditions	The vehicle controls all driving tasks under all conditions

History and present

- First examples in the 80s: Navlab 1 (1986), ALVINN (1989)…
- DARPA Grand Challenge (2004) and Urban Challenge (2007).
- Advanced driver-assistance systems (ADAS).

NAVLAB 1

WAYMO

NAYVA

WAYVE

STANLEY

CRUISE

TESLA

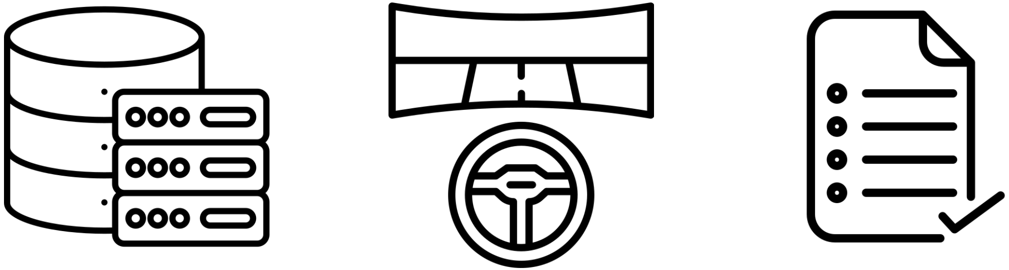
WAABI

Range of application

- Diverse range of environments and weather conditions.
- Urban scenarios, roads, highways.
- Use of maps.
- Vast range of vehicle types.
- Robustness is key (high-speed motion and passengers).

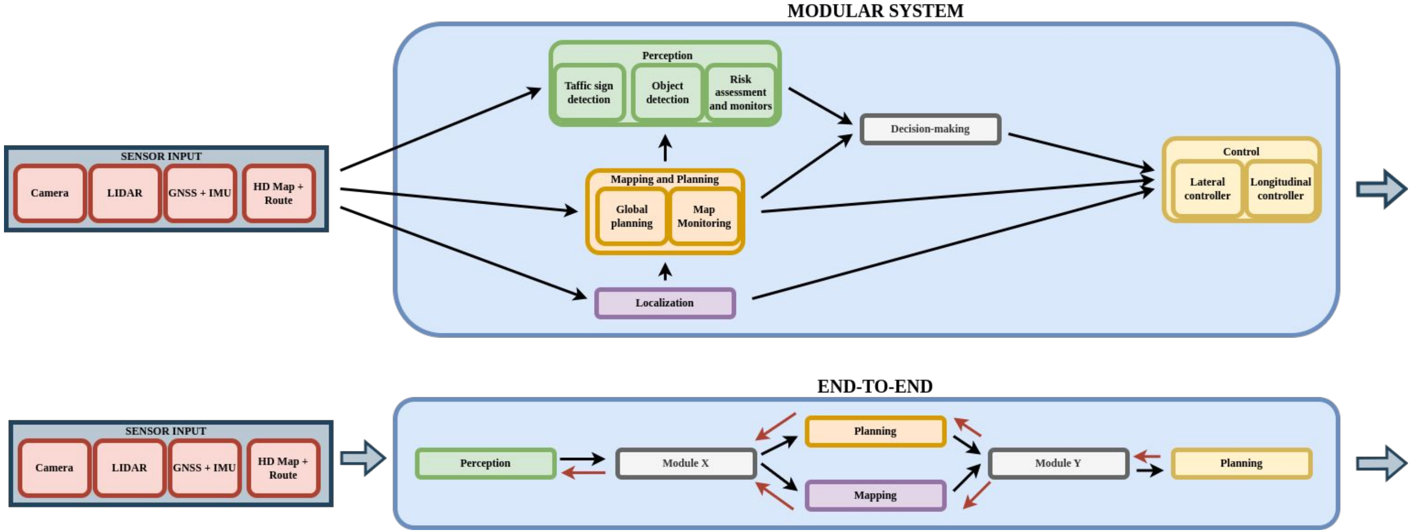
Assessment and metrics

- AD includes a broad spectrum of methodologies aimed at evaluating solutions derived from research.
- Studying the behavior carefully is crucial for understanding both its strengths and weaknesses.
- Massive datasets and simulators.
- Automatic and massive testing.
- Offline vs online.



End-to-end and modular approaches

- 2 main approaches in the development of AD solutions:
  - Modular.
  - **End-to-end (E2E).**
- The majority are currently modular but there is a trend towards end-to-end.



Research goals

Make contributions in the field of vision-based autonomous driving using deep learning, imitation learning and end-to-end models.

1. Deep learning-based visual traffic monitoring tool.
2. Generate software for validating object detection solutions.
3. Create software to assess autonomous driving behaviors and generate quantitative data.
4. Study of enhancements to visual lane following using visual memory and kinematic input.
5. Optimizing AD vehicle controllers to enhance performance.
6. Create and E2E shallow model for driving in traffic using vision.

Outline

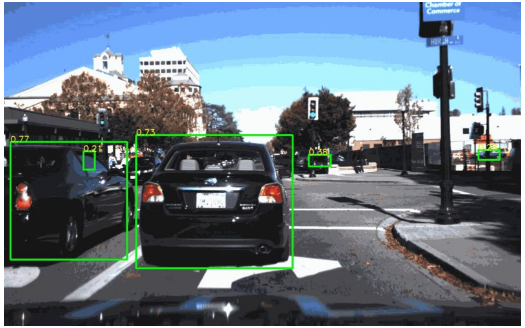
- Introduction
- **State-of-the-art**
- TrafficSensor
- DetectionMetrics
- Behavior Metrics
- Adding memory
- Optimizing DL models
- Driving in traffic
- Conclusions and future research

Traffic monitoring

- Traffic monitoring is a classic problem within CV.
- Traffic surveillance using video cameras was significantly improved with DL.

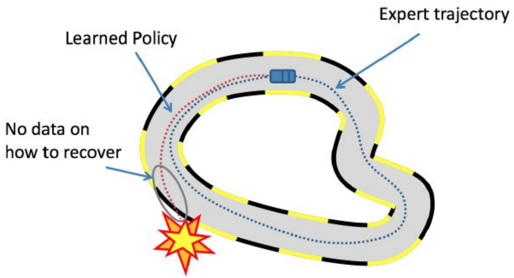
Deep learning object detection, datasets and assessment

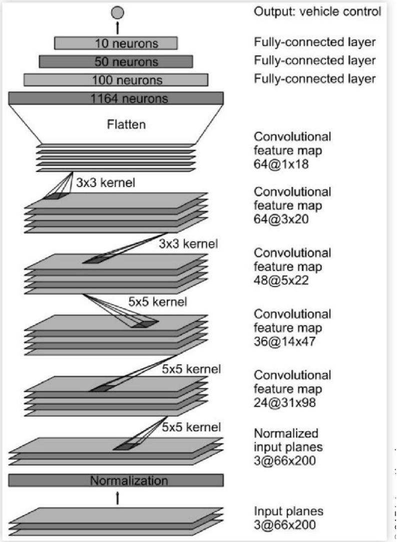
- Significant popularity lately. Notable advancements.
- Part of the perception in autonomous driving systems.
- Many datasets are available like COCO, ImageNet, Pascal VOC… and some others specialized.
- Each CV tasks includes a specific set of evaluation metrics.
- All the current state-of-the art solutions rely on DL.
- Faster R-CNN, SSD, YOLO, DETR, Swin Transformer, DINO…

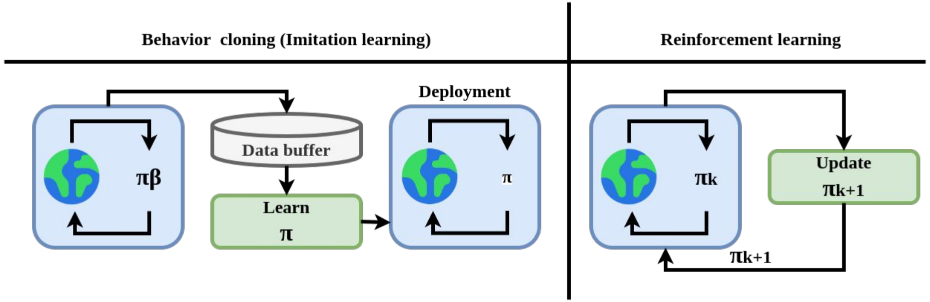


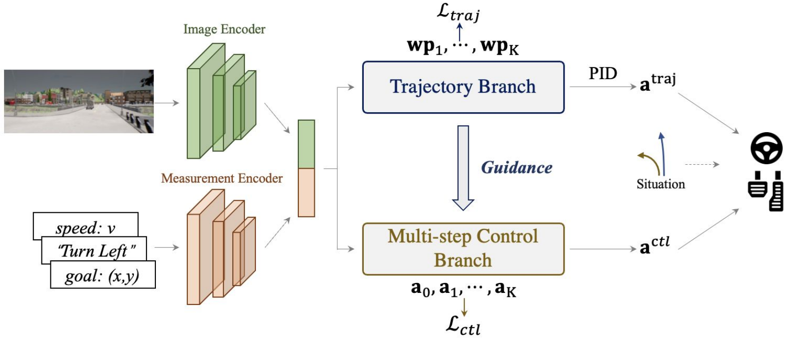
Imitation learning and RL for driving autonomously

- **Imitation learning** (behavior cloning).
- Reinforcement learning (RL- DRL).
- **PilotNet. Vision-based E2E AD**.
- TCP, ReasonNet, Transfuser…









Simulation in AD and assessment. Datasets

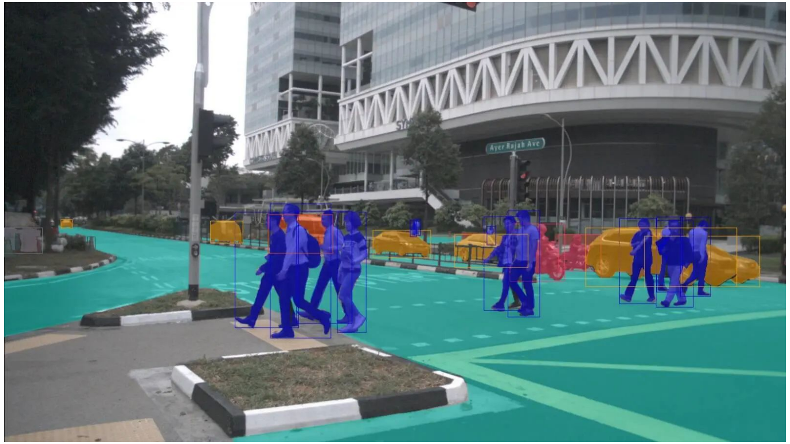
- Importance of simulation in robotics, in addition to real-world testing.
- Diverse simulators for AD are available: SUMO, TORCS, Gazebo, CARLA, DeepDrive, Autoware, proprietary simulators, world generators…

CARLA

Gazebo

Simulation in AD and assessment. Datasets

- In AD development, there are a lot of specialized datasets: nuScenes, BDD100K, KITTI, Cityscapes…
- CARLA Leaderboard for the assessment of solutions.
- Many different tasks and specific metrics needed.



Outline

- Introduction
- State-of-the-art
- **TrafficSensor**
- DetectionMetrics
- Behavior Metrics
- Adding memory
- Optimizing DL models
- Driving in traffic
- Conclusions and future research

Perception content block

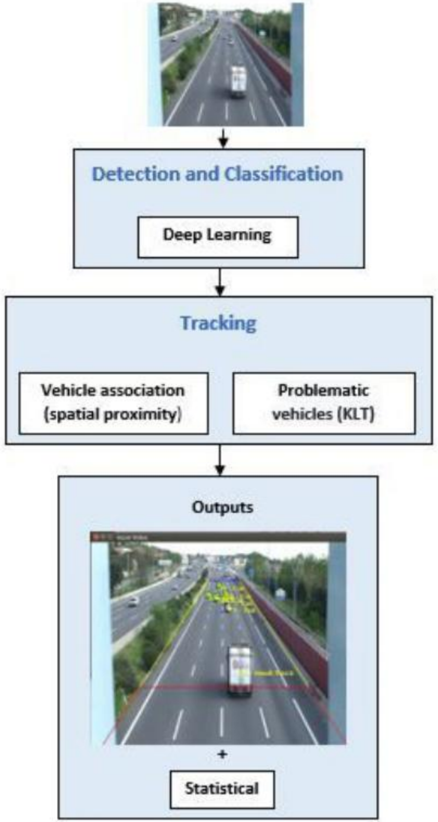
Monitoring and assessing traffic with deep learning

- Monitoring of real-time traffic on highways, road, and streets may provide useful data both for infrastructure planning and for traffic management in general.
- **TrafficSensor**: open-source system that employs deep learning for automatic vehicle tracking and classification on highways using a calibrated and fixed camera.
- Evolution of previous system (TrafficMonitor).
- Intro to traffic application domain, CV, AD.

J. Fernández, J. M. Cañas, V. Fernández, and S. Paniego, "Robust real-time traffic surveillance with deep learning," Computational Intelligence and Neuroscience, vol. 2021, p. 4 632 353, Dec. 2021. doi: 10.1155/2021/4632353. [Online]. Available: https://doi.org/10.1155/2021/4632353

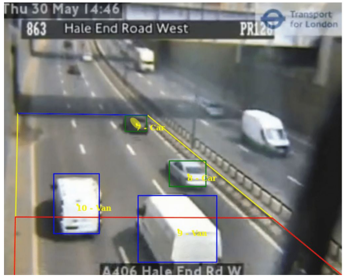
Monitoring and assessing traffic with deep learning

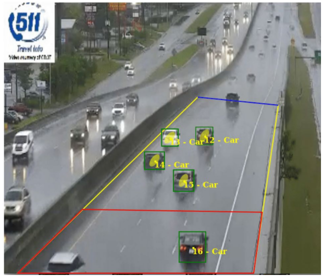
- New dataset created (7 vehicle types).
- Two modules:
  - Vehicle detection and classification.
  - Vehicle tracking.
- 1st module: YOLOv4.
- 2nd module: simple spatial association algorithm and more sophisticated KLT tracker to follow the vehicles.
- Several experiments validated the system.



Experimental evaluation

- System validated with a dataset of real traffic images.
- Detection Metrics for comparing different object detection DL models **(my contribution)**.
- Dataset (~10000 examples):
  - Good weather conditions.
  - Bad weather conditions.
  - Poor-quality images.





Conclusions

- TrafficSensor is a solution for vehicle surveillance using deep learning. 7 classes.
- Different DL models were studied and tested.
- A new dataset was curated for this application.
- The application proves to be robust to bad weather conditions, and blurred or low-resolution traffic images.
- State-of-the-art enhancements.

Outline

- Introduction
- State-of-the-art
- TrafficSensor
- **DetectionMetrics**
- Behavior Metrics
- Adding memory
- Optimizing DL models
- Driving in traffic
- Conclusions and future research

Assessing object detection deep learning architectures with quantitative metrics

- **Detection Metrics.**
- Comparison tool for the objective assessment of object detection approaches.
- Open-source.

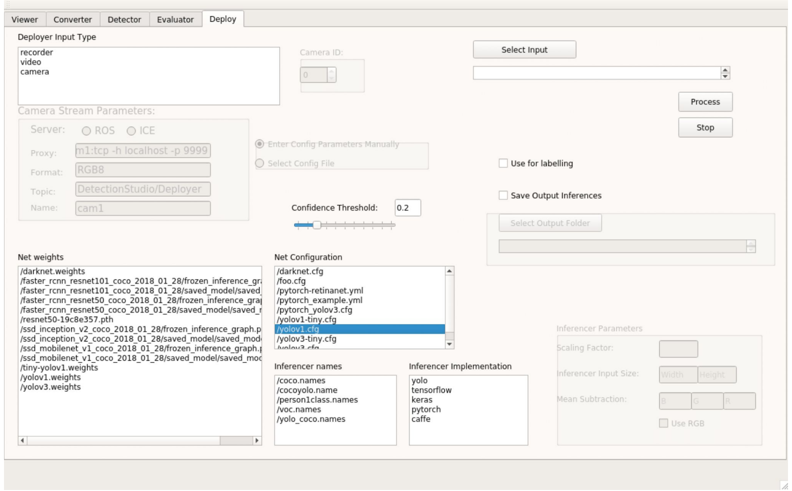
S. Paniego, V. Sharma, and J. M. Cañas, “Open source assessment of deep learning visual object detection,” Sensors, vol. 22, no. 12, 2022. doi: 10.3390/s22124575. [Online]. Available: https://www.mdpi.com/1424-8220/22/12/4575

Introduction

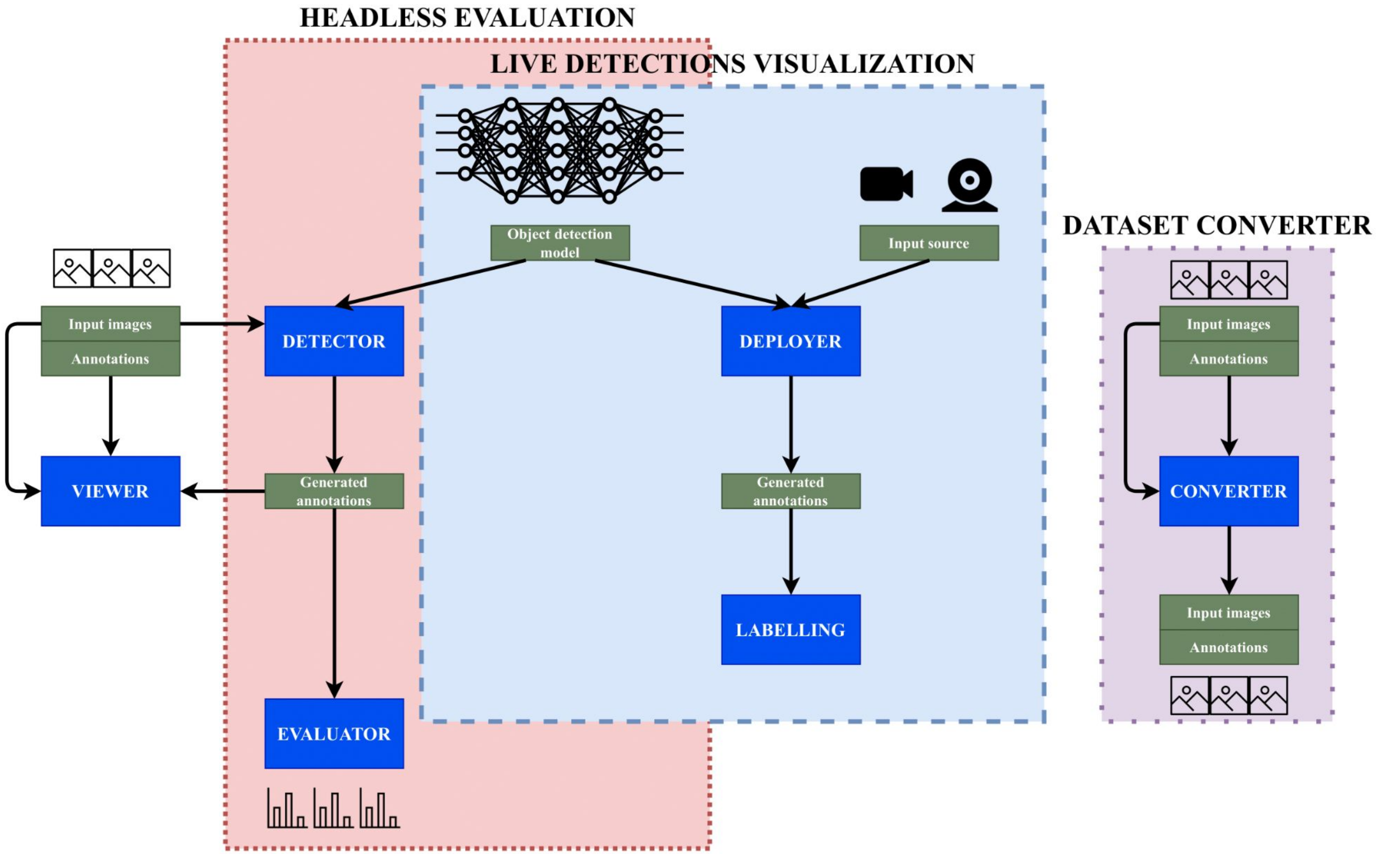
- Objective performance metrics.
- Systematic evaluation of different DL models for object detection on large datasets.
- Broad range of applicability: traffic monitoring, autonomous driving perception…
- Suite of tools with unique features to facilitate the objective comparison of different DL models for object detection.

Detection Metrics tool kit

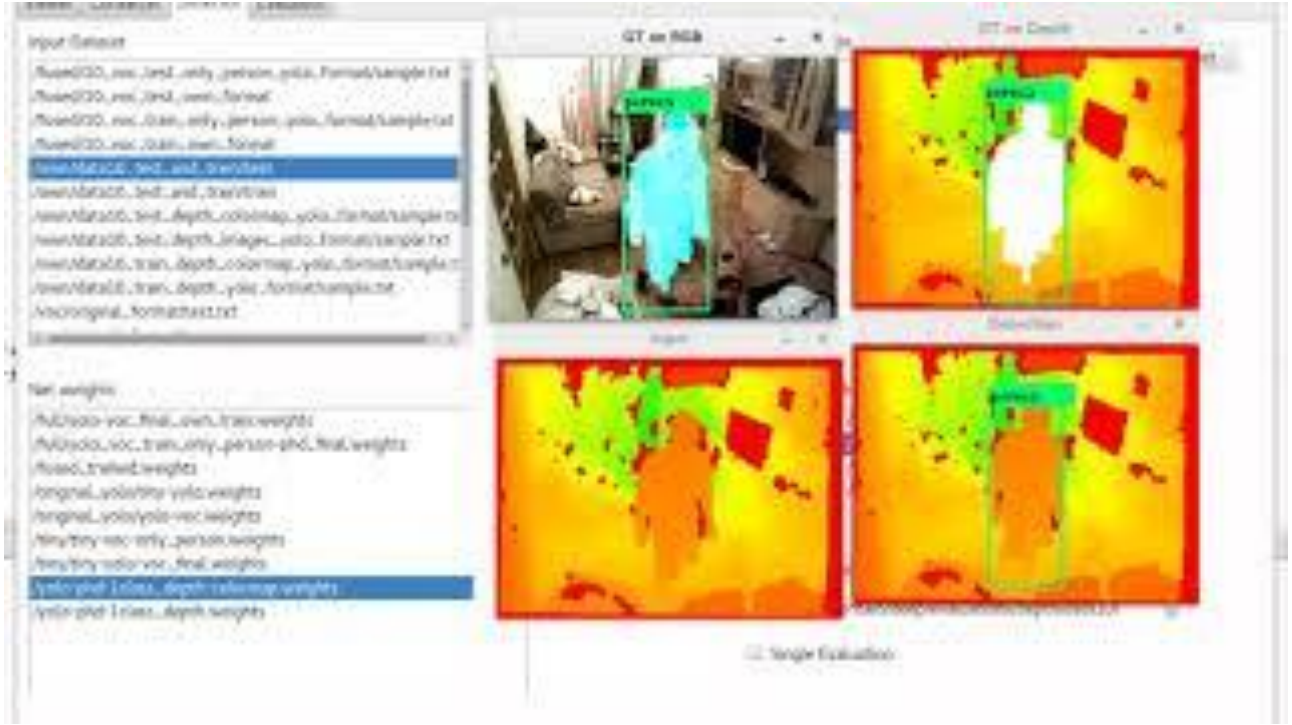
- DL frameworks support.
- Object detection datasets support.
- It can be used as a ROS Node.
- Two workflows:
  - Headless.
  - Graphical.
- Main tools:
  - Detection generation.
  - Evaluation of detections with objective metrics.
  - Live detection visualization.
  - Dataset converter.



Detection Metrics tool kit

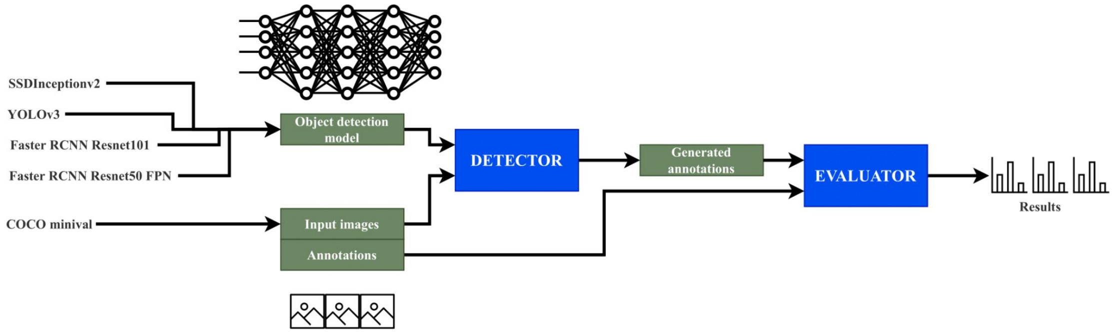


Detection Metrics tool kit



Experimental results and discussion

- Used in TrafficSensor validation.
- Comparison of state-of-the-art detection networks.



Network	Framework	Published mAP	mAP using Detection Metrics	Published mAR	mAR using Detection Metrics	Published Mean inference time	Mean inference time using Detection Metrics
SSD Inceptionv2	TensorFlow-Keras	0.24	0.27	✘	0.31	42	44
YOLOv3	Darknet	0.55 (IoU = 0.5)	0.47 (IoU = 0.5)	✘	0.5 (IoU = 0.5)	29	31
Faster RCNN Resnet101	TensorFlow-Keras	0.32	0.37	✘	0.43	106	122
Faster RCNN Resnet50 FPN	PyTorch	0.35	0.37	✘	0.46	59	102

Conclusions

- DetectionMetrics: open source assessment tool for developing object detection solutions validated in several experiments.
- State-of-the-art enhancements.

Outline

- Introduction
- State-of-the-art
- TrafficSensor
- DetectionMetrics
- **Behavior Metrics**
- Adding memory
- Optimizing DL models
- Driving in traffic
- Conclusions and future research

Control and decision-making content block

Assessing autonomous driving behaviors fine-grained metrics

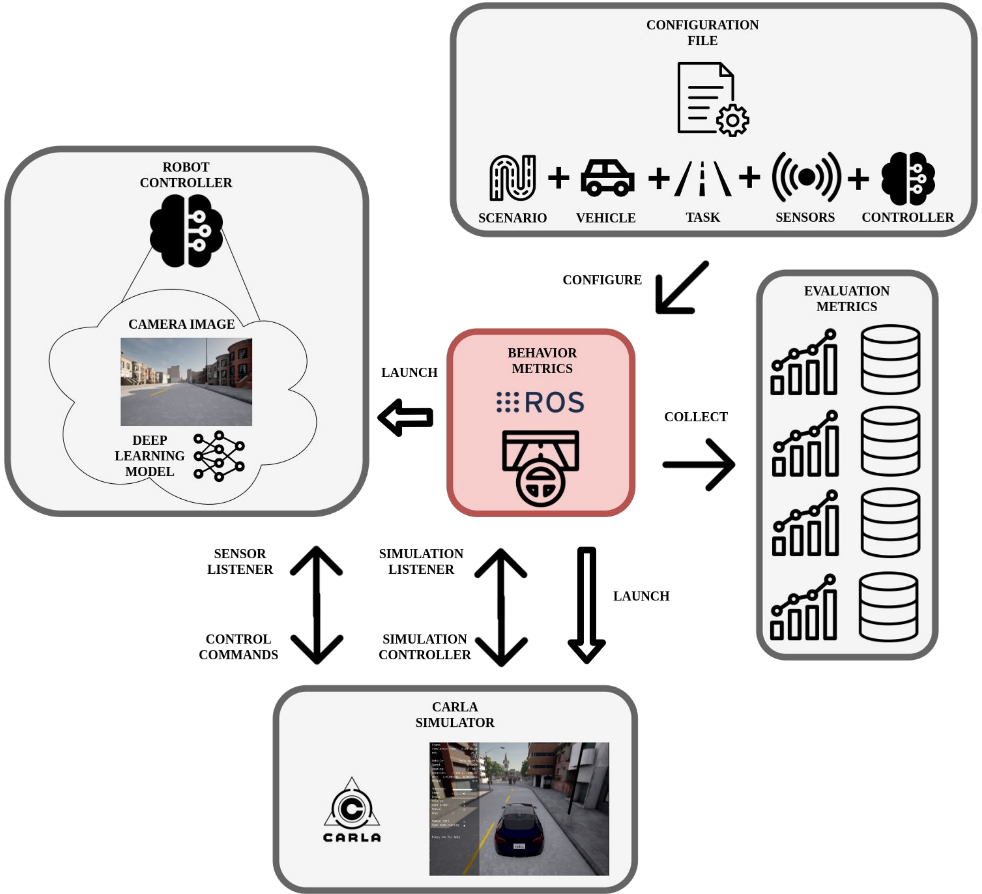
- **Behavior Metrics** is an open-source software tool developed to help research in the autonomous driving field.
- The development and validation of autonomous driving solutions require testing broadly in simulation (online).
- Quantitative and qualitative assessment and comparison of solutions for the main autonomous driving tasks.

S. Paniego, R. Calvo-Palomino, and J. Cañas, "Behavior Metrics: An Open-Source Assessment Tool for Autonomous Driving Tasks," Software X, vol. 26, pp. 101702, 2024 doi: 10.1016/j.softx.2024.101702. [Online]. Available: https://doi.org/10.1016/j.softx.2024.101702

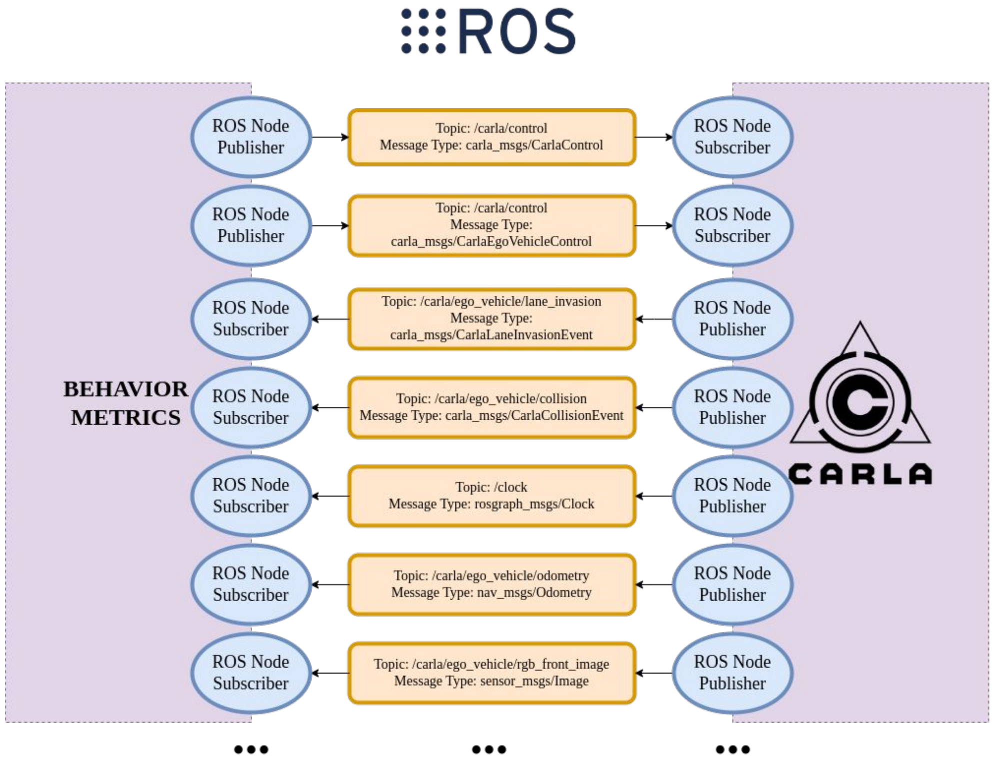
Assessing autonomous driving behaviors fine-grained metrics

- Two main evaluation pipelines:
  - Graphical.
  - Headless.
- Generates quantitative metrics complementary to the simulator's, including **fine-grained metrics** for each particular driving task (lane following, driving in traffic, route navigation…).
- It uses and supports Carla, ROS, PyTorch and Tensorflow.

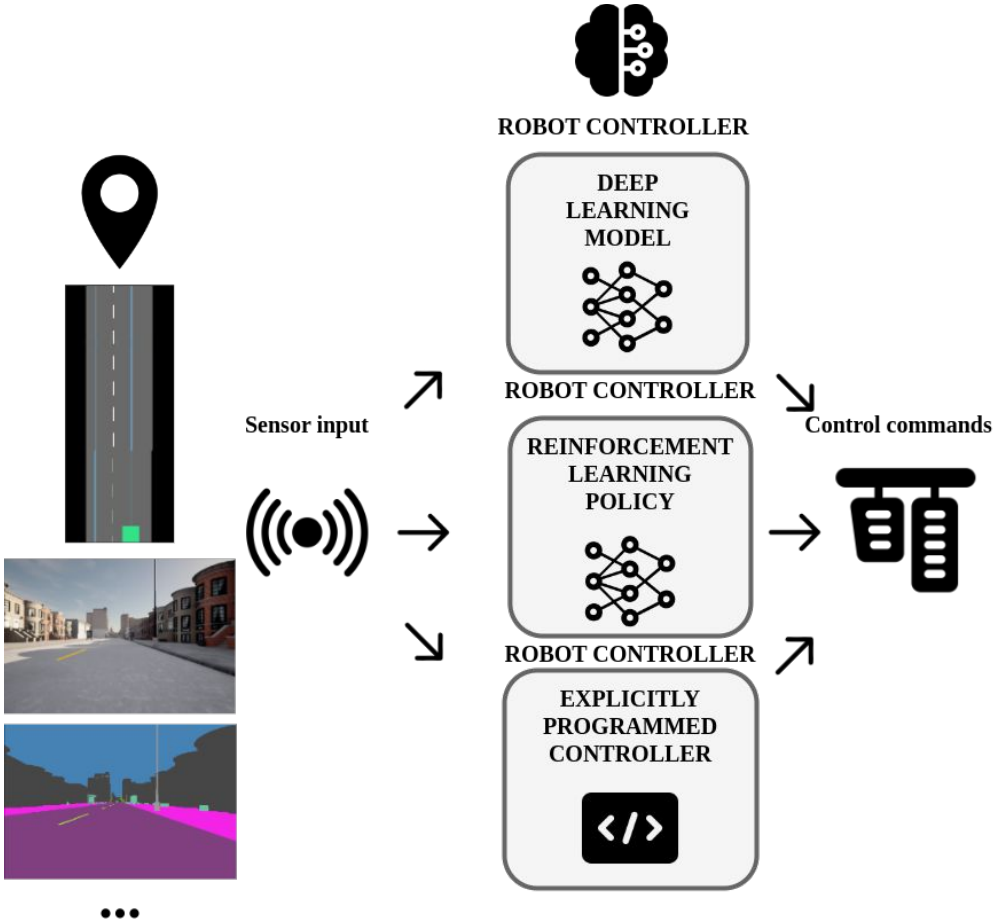
Assessing autonomous driving behaviors fine-grained metrics



Connection between BM and CARLA



Robot controller

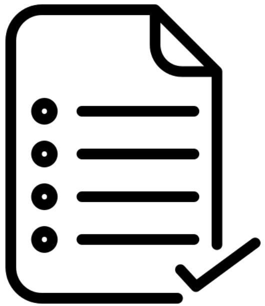


Supported tasks

- It currently supports three main tasks:
  - Lane following.
  - Driving in traffic.
  - Point-to-point navigation.

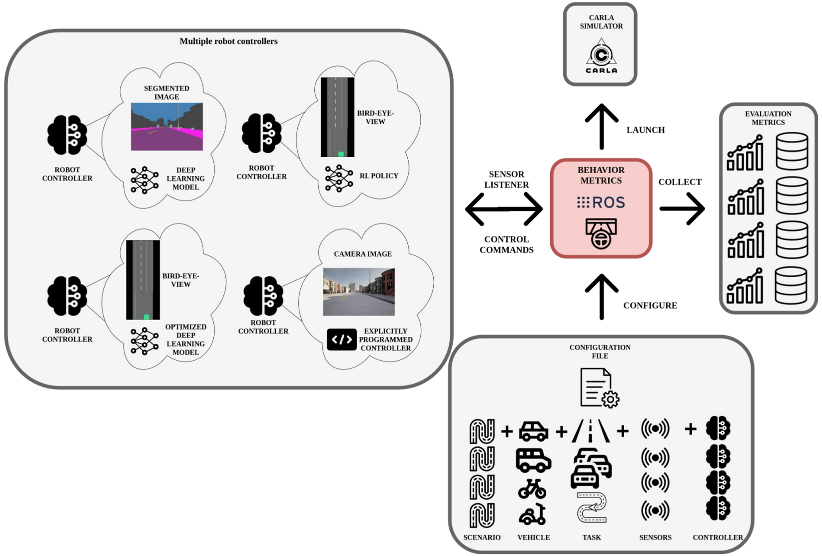
Autonomous driving evaluation metrics

- The proposed metrics are complementary to the simulator metrics (CARLA Leaderboard) and specific for each particular task:
  - Mean position deviation per km (MPD).
  - Effectively completed distance.
  - Vehicle longitudinal jerk per km.
  - Robot controller iteration frequency.
  - GPU inference frequency.
  - Collisions per km.
  - Lane invasions per km.
  - Distance to the front vehicle.
  - Route completion percentage.
  - Average speed.
  - Successful experiments.
  - …



GUI evaluation mode

Headless evaluation mode



Behavior Metrics headless mode

Impact and conclusions

- Used as an assessment tool in this thesis.
- Need for specialized assessment tool in AD for easy extraction of conclusions from experiments with fine-grained metrics.
- Open-source.

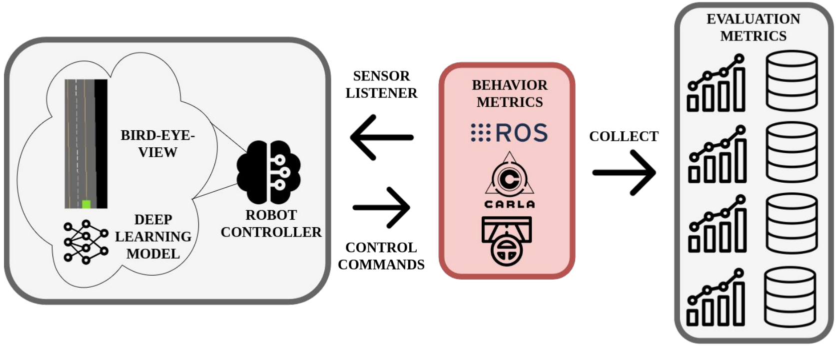
Outline

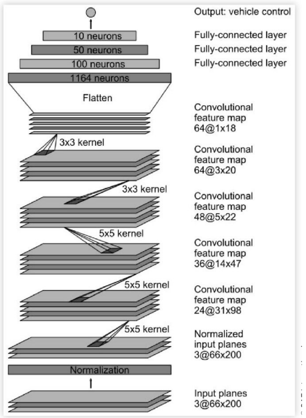
- Introduction
- State-of-the-art
- TrafficSensor
- DetectionMetrics
- Behavior Metrics
- **Adding memory**
- Optimizing DL models
- Driving in traffic
- Conclusions and future research

After tools development, solutions for imitation learning

Enhancing E2E AD control through kinematic input and memory-based architectures

- Baseline: imitation learning **follow lane** using PilotNet.
- **Explored and compared various approaches to enhance the capabilities of an E2E system for AD based on imitation learning adding visual memory and kinematic data input.**
- Offline and online evaluation (Behavior Metrics).

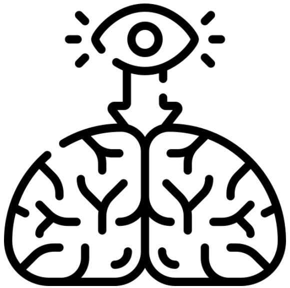




S. Paniego, R. Calvo-Palomino, and J. M. Cañas, "Enhancing End-to-End Control in Autonomous Driving through Kinematic-Infused and Visual Memory Imitation Learning," Manuscript submitted for publication, 2024.

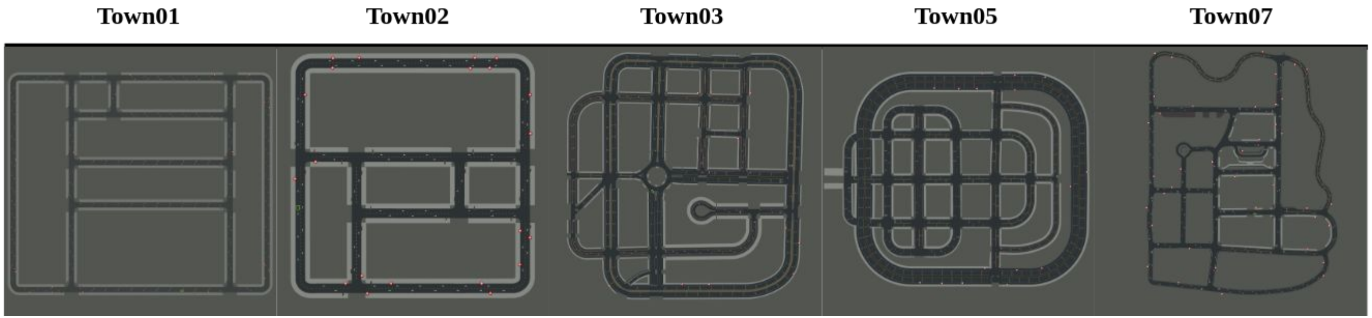
Enhancing E2E AD control through kinematic input and memory-based architectures

- Lane-following application using different urban scenario layouts and visual bird-eye-view input.
- Memory addition:
  - Architectural modifications.
  - Different types of sensory input.
- Incorporating visual memory and kinematic input: more robustness and wider range of challenging situations (reduction of collisions and speed self-regulation).
- LSTM, convLSTM…
- Open source materials.

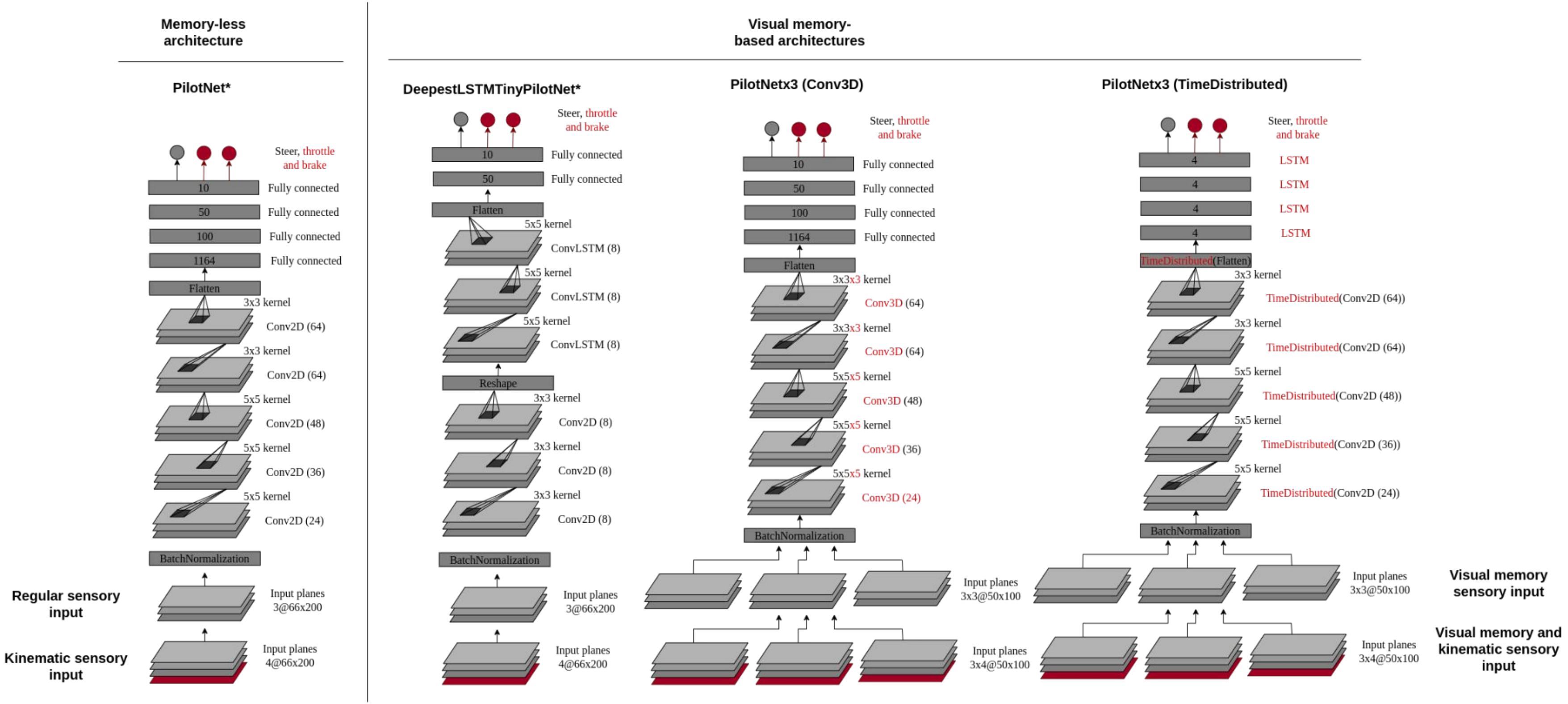


Kinematic-infused and visual memory E2E control based on imitation learning

- 8 DL models based on an E2E approach.
- 4 different DL architectures are explored:
  - One without memory.
  - Three with visual memory.
- PilotNet (Bojarski et al. 2016).
- Segmented bird-eye-view as input.
- Expert agent data.

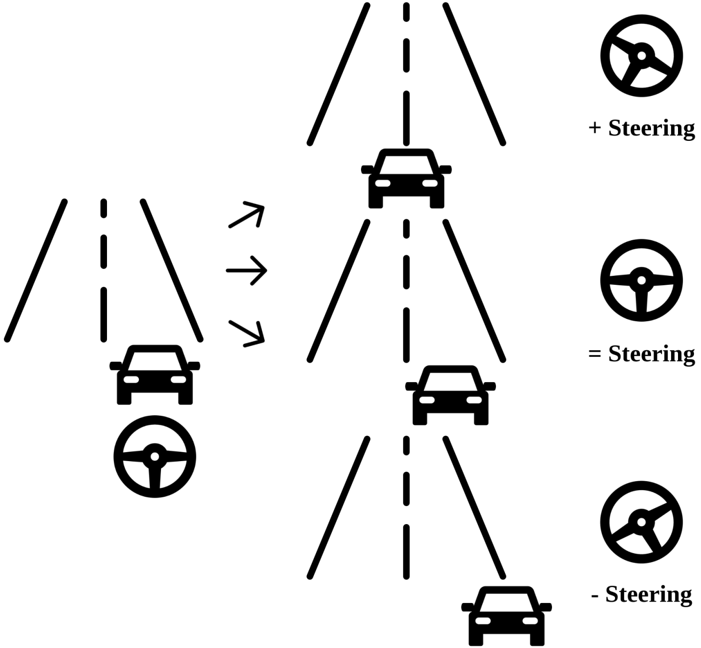


Kinematic-infused and visual memory E2E control based on imitation learning

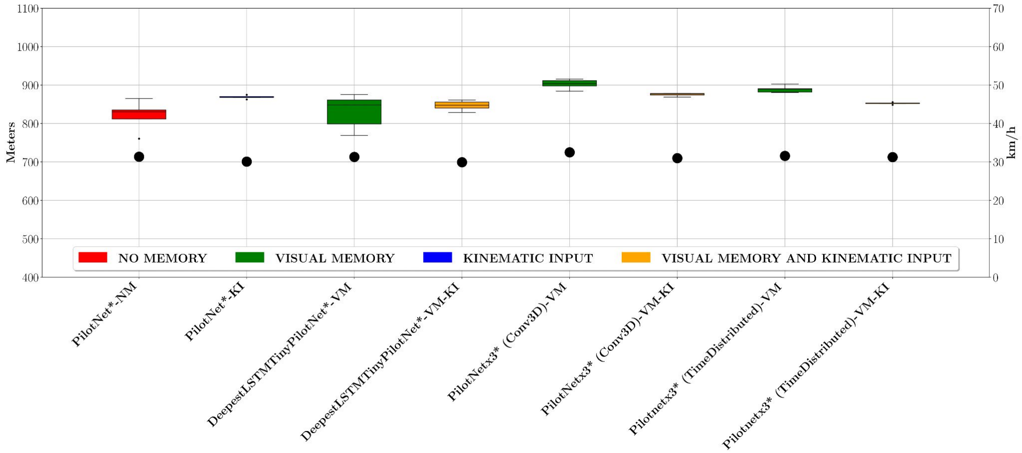


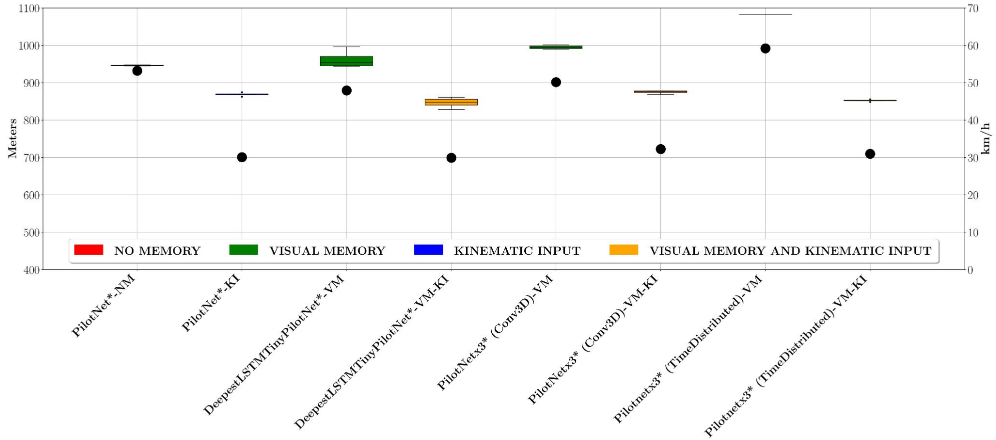
Kinematic-infused and visual memory E2E control based on imitation learning

- Training:
  - 20Hz for the collection.
  - Expert agent 30 km/h max.
  - Data augmentation. Affine.

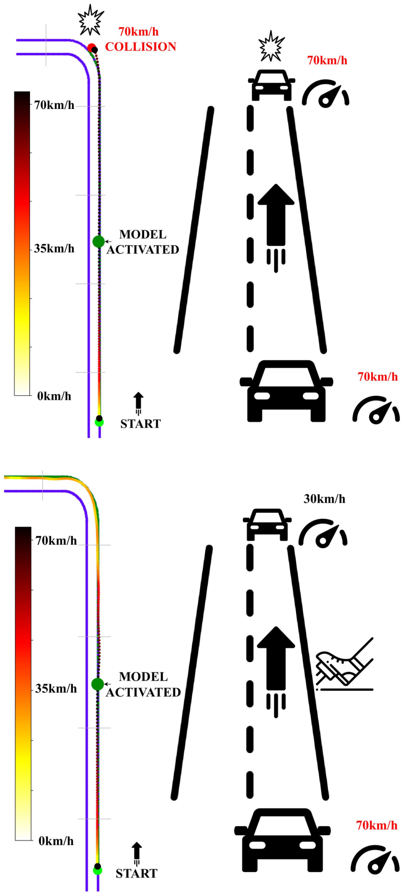


Top speed regulation vs no regulation





Taking control of a fast-moving car

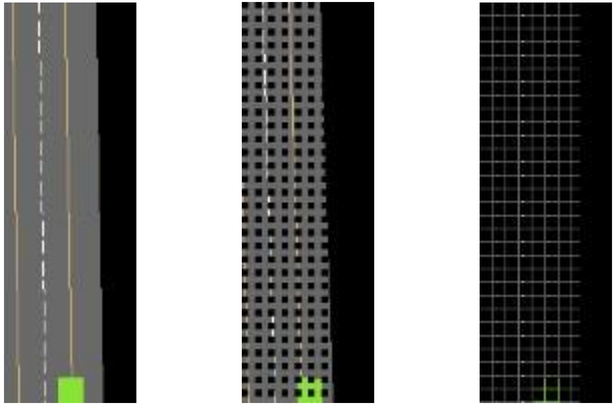


Map	Town02			
Model	Pilotnet*	DeepestLSTMTinyPilotNet*	Pilotnetx3* (Conv3D)	Pilotnetx3* (TimeDistributed)
Visual memory	✘	✔	✔	✔
Kinematic input	✔	✔	✔	✔
Speed	50 km/h			
Experiments with collisions	5/5	0/5	0/5	0/5
Average speed	-	27.18	27.07	26.82
Collisions per km	46.51	0.0	0.0	0.0
Successful experiments	0/5	**5/5**	**5/5**	**5/5**
Speed	70 km/h			
Experiments with collisions	5/5	5/5	0/5	5/5
Average speed	-	-	29.92	-
Collisions per km	27.25	29.95	0.0	26.95
Successful experiments	0/5	0/5	**5/5**	0/5

Table 6.4: Comparison of models in a high-speed scenario where the model takes control when the ego vehicle is already at a speed of 70 km/h. For the *Average speed*, we only consider experiments without collisions. This experiment is tested in Town02. Values in **bold** highlight the most interesting results. ✔: supported. ✘: unsupported.

Robustness to sensory manipulation

50% 90%



Map	Town02			
Model	Pilotnet*	DeepestLSTMTinyPilotNet*	Pilotnetx3* (Conv3D)	Pilotnetx3* (TimeDistributed)
Visual memory	✘	✔	✔	✔
Kinematic input	✔	✔	✔	✔
Percentage	50%			
Experiments with collisions	0/5	1/5	5/5	0/5
Average speed	26.12	22.54	-	25.96
Collisions per km	0.0	1.52	72.75	0.0
Successful experiments	**5/5**	0/5	0/5	**5/5**
Percentage	90%			
Experiments with collisions	5/5	0/5	5/5	0/5
Average speed	-	5.19	-	25.18
Collisions per km	23.68	0.0	12.31	0.0
Successful experiments	0/5	0/5	0/5	**5/5**

Table 6.5: Comparison of model performance modifying the input sensory information. For the *Average speed*, we only consider experiments without collisions. Values in **bold** highlight the most interesting results. ✔: supported. ✘: unsupported.

Visual memory length and density

Map	Town02					
Model	Pilotnetx3* (Conv3D)			Pilotnetx3* (TimeDistributed)		
Visual memory	✔	✔	✔	✔	✔	✔
Kinematic input	✘	✘	✘	✘	✘	✘
Memory length (frames)	3	5	9	3	5	9
Collisions	0	0	1.0	0	0.8	1.0
Average speed	25.18	25.64	-	26.08	26.65	-
Position deviation mean per km	1.13	1.75	-	1.12	2.02	-
Collisions per km	0.0	0.0	1.30	0.0	0.0	6.25
Successful experiments	**5/5**	**5/5**	0/5	**5/5**	1/5	0/5

Table 6.6: Comparison of model performance with different visual memory lengths. For the *Average speed* and *Position deviation mean per km*, we only consider experiments without collisions. Values in **bold** highlight the most interesting results. ✔: supported. ✘: unsupported.

Map	Town02			
Model	Pilotnetx3* (Conv3D)			
Visual memory	✔	✔	✔	✔
Kinematic input	✘	✘	✘	✘
Memory densities (frames)	$t, t-1, t-2$	$t, t-5, t-10$	$t, t-10, t-20$	$t, t-20, t-40$
Collisions	0.0	0.0	0.4	0.6
Average speed	24.96	25.18	26.59	26.07
Positions deviation mean per km (m/km)	1.39	1.13	1.35	1.89
Collisions per km	0.0	0.0	5.13	5.12
Successful experiments	**5/5**	**5/5**	3/5	2/5
Model	Pilotnetx3* (TimeDistributed)			
Visual memory	✔	✔	✔	✔
Kinematic input	✘	✘	✘	✘
Memory densities (frames)	$t, t-1, t-2$	$t, t-5, t-10$	$t, t-10, t-20$	$t, t-20, t-40$
Collisions	0.0	0.0	0.2	0.2
Average speed	25.94	26.08	26.63	26.01
Positions deviation mean per km (m/km)	1.12	1.12	1.47	1.59
Collisions per km	0.0	0.0	2.97	3.01
Successful experiments	**5/5**	**5/5**	4/5	4/5

Table 6.7: Comparison of model performance with different visual memory densities. For the *Average speed* and *Position deviation mean per km*, we only consider experiments without collisions. Values in **bold** highlight the most interesting results. ✔: supported. ✘: unsupported.

Conclusions

- Adding visual memory and kinematic input to the models enhances the quality of the final control following the lane.
- Adding kinematic data, the system controls the speed better.
- Combining it with visu … its range of application.

ExperimentType	Visual	Visual memory	Kinematic input	Visual memory and kinematic input
Regular lane-follow (Section 6.4.2)	✔	✔	✔	✔
Experiment without top speed (Section 6.4.3)	✘	✘	✔	✔
High speed experiment (Section 6.4.4)	✘	✘	✘	✔
Sensory Robustness experiments (Section 6.4.5)	✘	✘	✘	✔

Table 6.8: Comparison summary of model performance across presented experiments. The addition of at least kinematic input data improves the final behavior and adding both types generates gains in certain scenarios. ✔: successful. ✘: failure.

Outline

- Introduction
- State-of-the-art
- TrafficSensor
- DetectionMetrics
- Behavior Metrics
- Adding memory
- **Optimizing DL models**
- Driving in traffic
- Conclusions and future research

Optimization of end-to-end autonomous driving control

- **We explore and compare a variety of alternatives for model optimization in the context of E2E imitation learning autonomous driving.**
- Performance depends on quality of decisions and their frequency. Increase frequency without sacrificing decision quality.
- Run deep learning robot-control application even in limited computed hardware.
- Open source materials.

Google Summer of Code

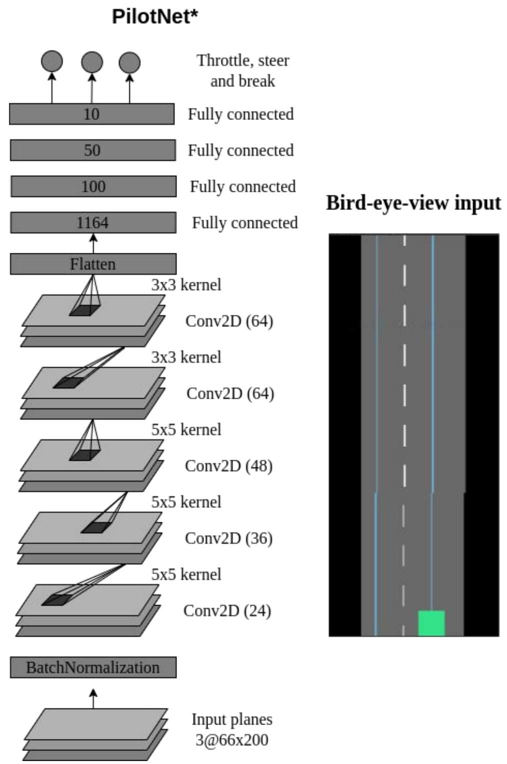
S. Paniego, N. Paliwal, and J. Cañas, “Model optimization in deep learning based robot control for autonomous driving,” IEEE Robotics and Automation Letters and IEEE International Conference on Robotics and Automation (ICRA), vol. 9, no. 1, pp. 715–722, 2024. doi: 10.1109/LRA.2023.3336244. [Online]. Available: https://doi.org/10.1109/LRA.2023.3336244

Optimization of end-to-end autonomous driving control

- Trend towards E2E solutions.
- DL models are high-demanding.
- Available computing hardware is key for performance.
- Some autonomous robots have high-performant hardware while others do not and is not always possible to upgrade (edge devices). 
- Optimization solution (framework support).

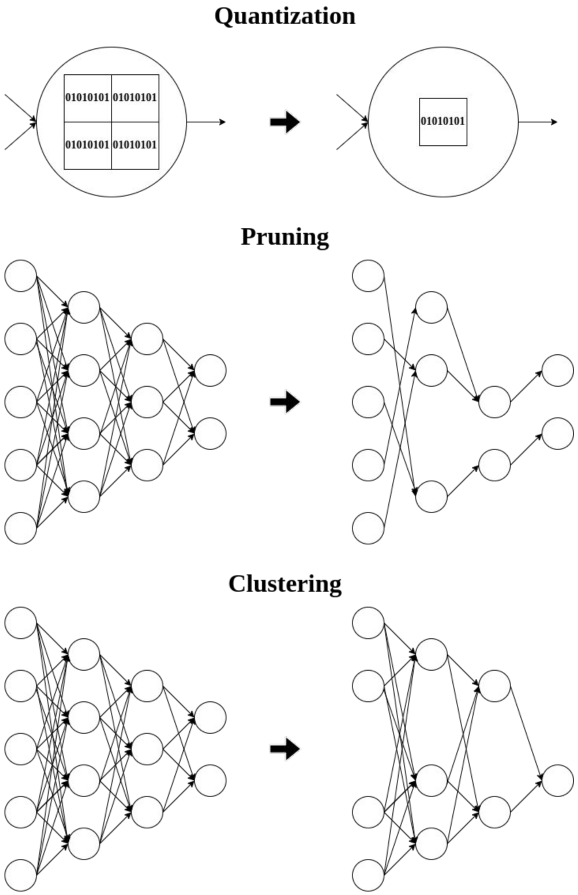
Optimizing E2E IL models for lane-follow robot control

- The baseline architecture is based on PilotNet (Bojarski et al. 2016), named PilotNet*.



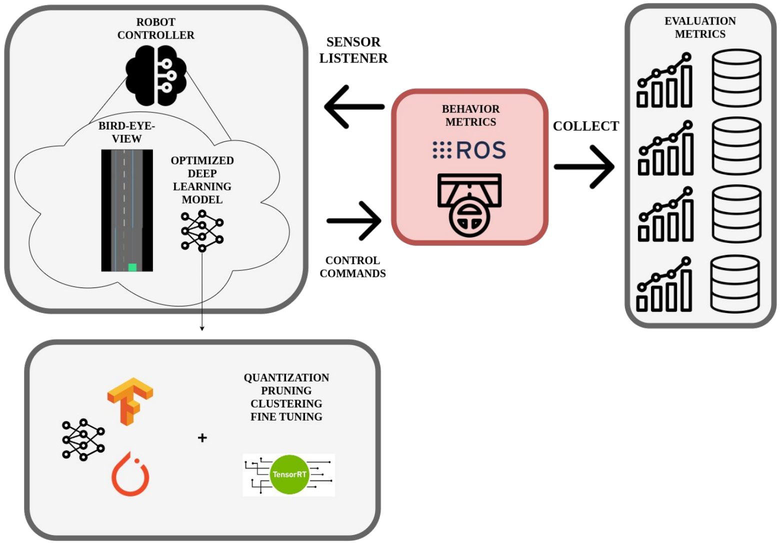
Optimizing E2E IL models for lane-follow robot control

- Optimizations used:
  - Quantization (float16, int8).
  - Pruning.
  - Fine tuning (retraining).
  - Clustering.



Assessment of Models

- Behavior Metrics.
- In addition to the simulator's metrics:
    - Controller frequency.
    - GPU inference frequency.
    - Position deviation mean per km.
    - Successful runs.

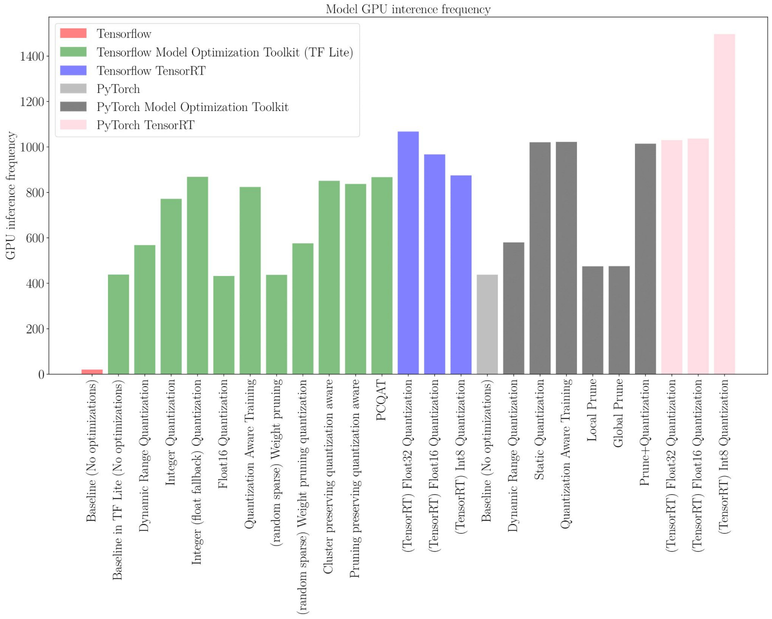


Experiment 1: Offline evaluation

- Model size reduction (12x int8 quantization).
- GPU inference frequency improves.
  - 135x TensorRT int8 quantization.
  - 50x framework-specific int8 quantization.
- Similar MSE.
- Combination of optimizations generates best case: PCQAT (Sparsity and cluster preserving quantization aware training).

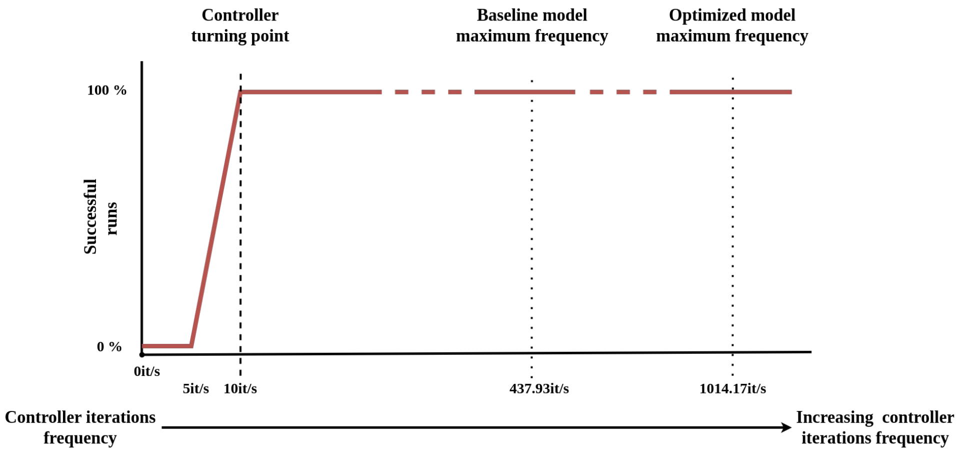
Experiment 2: Online evaluation

- Higher controller frequency.
- Faster models: int8 (47x TF, 2.3x PyTorch).
- Similar mean position deviation and average speed.
- Previous insights are applicable.
- No decision quality degradation in online evaluation.
- Faster and smaller models.
- TF and PyTorch similar. TensorRT best.
- Optimization limits require tuning.



Experiment 3: Inference freq. and Quality of Decisions in Robot Control Performance

- Quality changes when the frequency of decisions is changed in the robot control.
- Optimizing benefits control quality since in limited hardware, it may be critical to generate decisions above the turning point.



Conclusions

- Optimizations improve the system performance (controller speed-up without losing decision quality).
- Offline and online evaluation.
- Optimized models have wider range of application (edge devices).
- Combination of optimizations and hardware-specific optimizations generate the best models.

Outline

End-to-end vision-based autonomous driving in traffic

- **A shallow E2E vision-based DL approach for autonomous driving in traffic scenarios.**
- Goals: follow the lane and maintain a safe distance from possible preceding vehicles.
- We prioritize simplicity of the model.
- **Subjective visual input.**

S. Paniego, E. Sinohara, and J. M. Cañas, "Autonomous Driving in Traffic with End-to-End Vision-based Deep Learning," Neurocomputing, 2024. doi: 10.1016/j.neucom.2024.127874. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231224006453

Imitation learning for driving in traffic

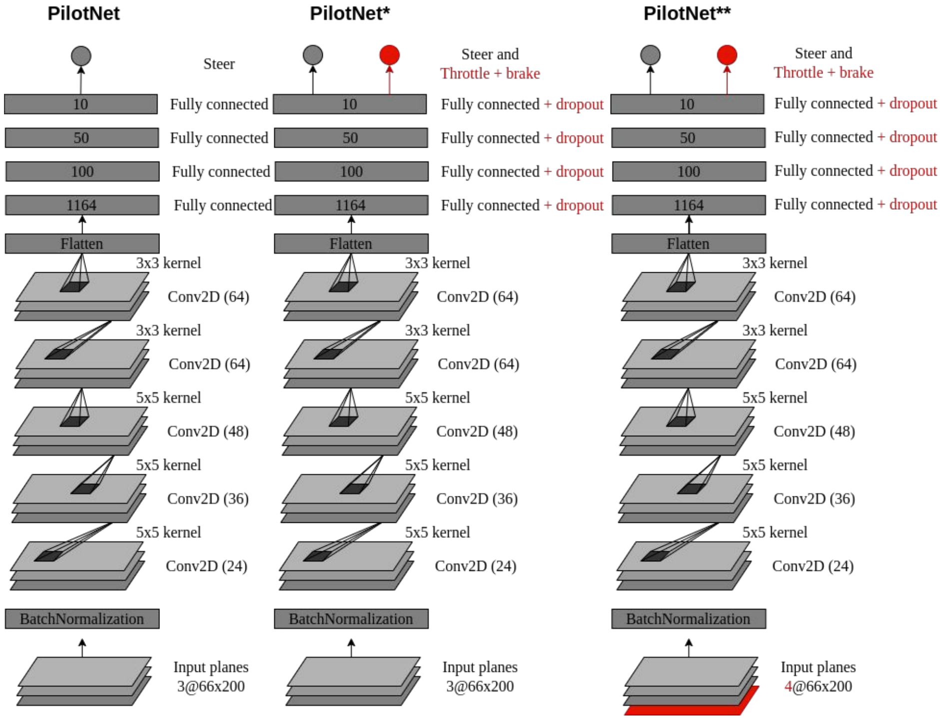
- Dataset extracted from expert agent behavior.
  - **Traffic-0:** does not consider traffic.
  - **Traffic-1:** includes one type of front vehicle.
  - **Traffic-6:** includes 5 other vehicles including vans and three urban cars of different sizes and colors.

Traffic-1

Traffic-6

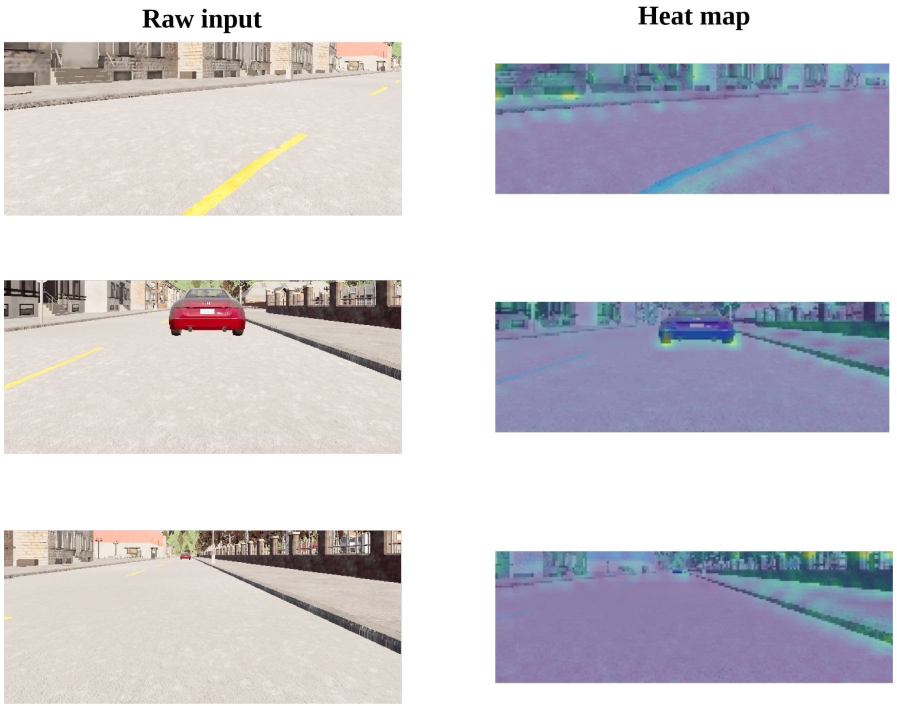
Imitation learning for driving in traffic

- 2 models are variations of PilotNet (**PilotNet*** and **PilotNet****).
- We include oversampling during training.



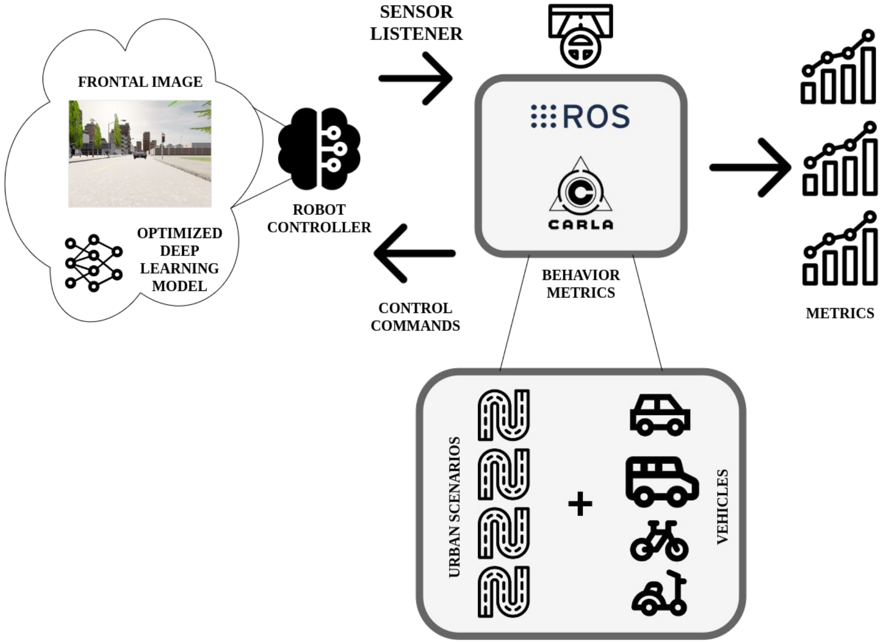
End-to-end vision-based autonomous driving in traffic

- We utilize activation heatmaps to understand where the models is paying attention during the decision making **(Grad-CAM)**.



Experimental setup

- We use Behavior Metrics for the online evaluation, including new specialized metrics (*distance to vehicle*).
- With small changes to the architecture of PilotNet, it is possible to expand its functionality.



Experiment 1: Execution without traffic

- All the combinations of model+dataset works correctly. The model trained with Traffic-6 is better in MPD and Lane invasions.

Table 8.2: Metrics for two different towns and models in free-road conditions. Success rate: the higher the better; the rest: the lower the better.

	Town01			Town02		
Model	**PilotNet***	**PilotNet****	**PilotNet****	**PilotNet***	**PilotNet****	**PilotNet****
Training dataset	**Traffic-1**	**Traffic-1**	**Traffic-6**	**Traffic-1**	**Traffic-1**	**Traffic-6**
Success Rate (%)	100	100	100	100	100	100
MPD	0.33	0.3	**0.19**	0.84	0.49	**0.32**
Lane Invasions	14.884	10.02	**4.75**	26.56	15.4	**3.42**

Experiment 2: Execution with traffic

- The model trained with Traffic-6 is better for all the metrics.

Table 8.3: Metrics for two different towns and models in in-traffic conditions

	Town01			Town02		
Model	**PilotNet***	**PilotNet****	**PilotNet****	**PilotNet***	**PilotNet****	**PilotNet****
Training dataset	**Traffic-1**	**Traffic-1**	**Traffic-6**	**Traffic-1**	**Traffic-1**	**Traffic-6**
Success Rate (%)	0	16	**81**	0	83	**100**
MPD	43.12	18.07	**0.26**	50.57	1.97	**0.32**
Lane Invasions	28.87	25.84	**6.54**	69.65	21.15	**1.48**

Table 8.4: Metrics for the distance to the front vehicle.

	Town01	
Model	**PilotNet****	**PilotNet****
Training dataset	**Traffic-1**	**Traffic-6**
Dangerous distance	6%	**2%**
Short distance	25%	16%
Medium distance	27%	30%
Great distance	42%	52%
Success rate	16%	**86%**

Experiment 3: generalization

- The model trained with Traffic-6 generalizes for a whole set of vehicles never-seen during training.

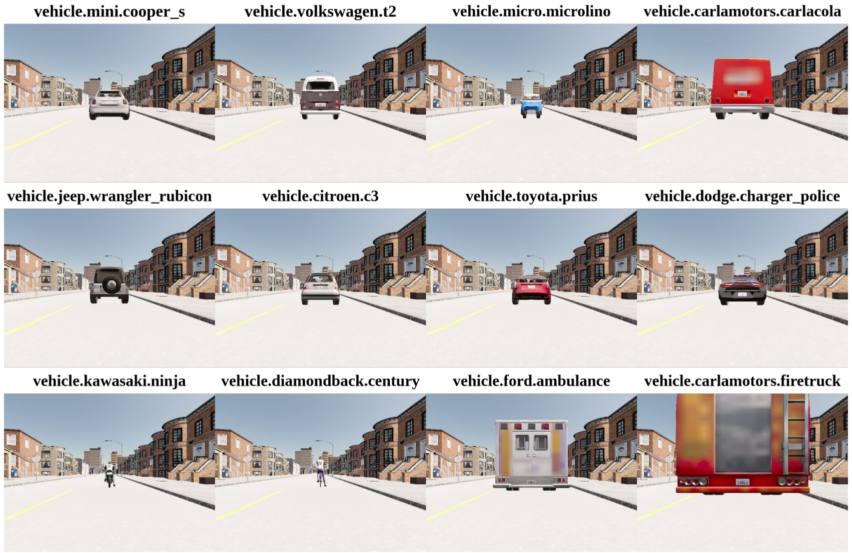


Table 8.5: Success rate metric for each of the 12 vehicles.

	Town01	
Model	**PilotNet****	**PilotNet****
Training dataset	**Traffic-1**	**Traffic-6**
vehicle.mini.cooper_s	50%	83%
vehicle.volkswagen.t2	0%	100%
vehicle.micro.microlino	50%	100%
vehicle.carlamotors.carlacola	0%	100%
vehicle.jeep.wrangler_rubicon	0%	100%
vehicle.citroen.c3	17%	100%
vehicle.toyota.prius	0%	83%
vehicle.dodge.charger_police	33%	83%
vehicle.kawasaki.ninja	0%	100%
vehicle.diamondback.century	0%	50%
vehicle.ford.ambulance	50%	66%
vehicle.carlamotors.firetruck	0%	66%

End-to-end vision-based autonomous driving in traffic

Conclusions

- Proposal for safe autonomous driving in traffic scenarios following and E2E vision-based approach with imitation learning and deep learning.
- The shallow models slightly changed can drive in traffic situations successfully as proved experimentally.
- We also include a new metrics to Behavior Metrics.
- TFM co-direction.

Outline

- Introduction
- State-of-the-art
- TrafficSensor
- DetectionMetrics
- Behavior Metrics
- Adding memory
- Optimizing DL models
- Driving in traffic
- **Conclusions and future research**

Conclusions and future research

Make contributions in the field of vision-based autonomous driving using deep learning, imitation learning and end-to-end models.

- ✓ Deep learning-based visual traffic monitoring tool.
- ✓ Generate software for validating object detection solutions.
- ✓ Create software to assess autonomous driving behaviors and generate quantitative data.
- ✓ Study of enhancements to visual lane following using visual memory and kinematic input.
- ✓ Optimizing AD vehicle controllers to enhance performance.
- ✓ Create and E2E shallow model for driving in traffic using vision.

Research contributions

- **5 journal and conference papers**:
  - S. Paniego, R. Calvo-Palomino, and J. Cañas, "Behavior Metrics: An Open-Source Assessment Tool for Autonomous Driving Tasks," Software X, vol. 26, pp. 101702, 2024 doi: 10.1016/j.softx.2024.101702. [Online]. Available: https://doi.org/10.1016/j.softx.2024.101702 [239]
  - S. Paniego, N. Paliwal, and J. Cañas, "Model optimization in deep learning based robot control for autonomous driving," IEEE Robotics and Automation Letters and IEEE International Conference on Robotics and Automation (ICRA), vol. 9, no. 1, pp. 715–722, 2024. doi: 10.1109/LRA.2023.3336244. [Online]. Available: https://doi.org/10.1109/LRA.2023.3336244 [244]
  - S. Paniego, V. Sharma, and J. M. Cañas, "Open source assessment of deep learning visual object detection," Sensors, vol. 22, no. 12, 2022. doi: 10.3390/s22124575. [Online]. Available: https://www.mdpi.com/1424-8220/22/12/4575 [234]
  - J. Fernández, J. M. Cañas, V. Fernández, and S. Paniego, "Robust real-time traffic surveillance with deep learning," Computational Intelligence and Neuroscience, vol. 2021, p. 4 632 353, Dec. 2021. doi: 10.1155/2021/4632353. [Online]. Available: https://doi.org/10.1155/2021/4632353 [224]
  - S. Paniego, E. Sinohara, and J. M. Cañas, "Autonomous Driving in Traffic with End-to-End Vision-based Deep Learning," Neurocomputing, 2024. doi: 10.1016/j.neucom.2024.127874. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231224006453

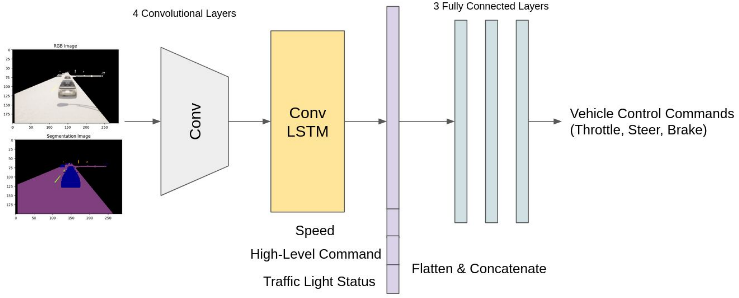
Research contributions

- **1 manuscript under peer-review:**
  - Sergio Paniego, Roberto Calvo-Palomino, and José María Cañas, “Enhancing End-to-End Control in Autonomous Driving through Kinematic-Infused and Visual Memory Imitation Learning”.
- **1 preprint:**
  - S. P. Blanco, S. Mahna, U. A. Mishra, and J. Canas, Memory based neural networks for end-to-end autonomous driving, 2022. arXiv: 2205.12124 [cs.RO]. [256]
- **1 workshop paper:**
  - P. F. de Cabo, R. Lucas, I. Arranz, S. Paniego, and J. M. Cañas, “RL-studio: A tool for reinforcement learning methods in robotics,” in ROBOT2022: Fifth Iberian Robotics Conference, Springer International Publishing, Nov. 2022, pp. 502–513.doi: 10.1007/978-3-031-21062-4_41. [Online]. Available: https://doi.org/10.1007%2F978-3-031-21062-4_41. [257]

Future (current) work 1

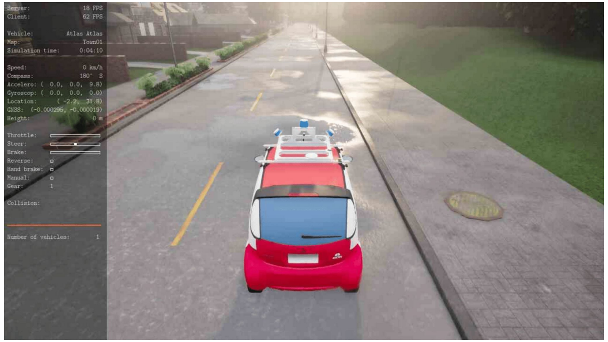
- Point-to-point E2E navigation using *input commands*.
- GSoC project.

Google
Summer of Code



Future (current) work 2

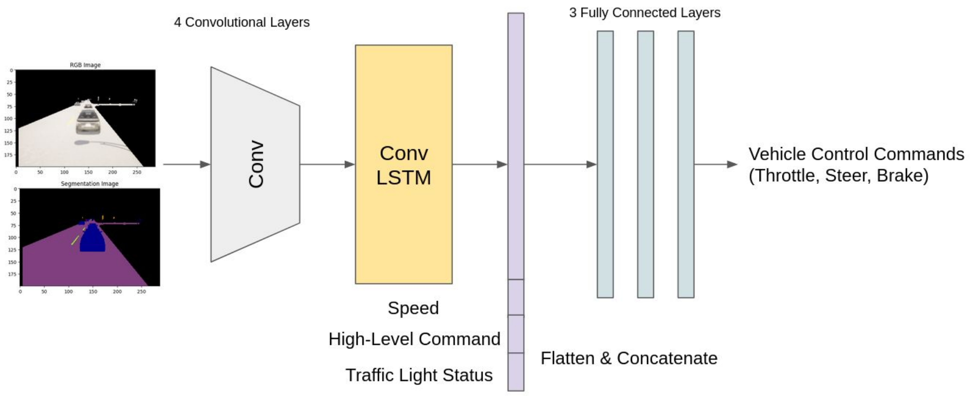
- Transferring current E2E solutions to a real-world vehicle.



Future (current) work 3

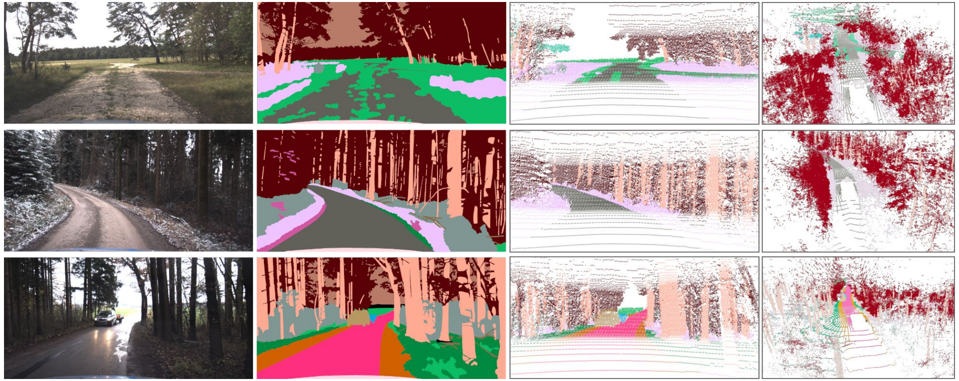
- E2E autonomous driving modulated with text-based instructions (LLMs).
- GSoC project.

Google Summer of Code



Future (current) work 4, 5, 6

- Exploration of E2E autonomous driving in unstructured environments.
- TFM co-direction.



- Exploration of E2E autonomous driving in aerial vehicles.

- Combination of driving experts.
- TFG co-direction.

Materials

Universidad
Rey Juan Carlos

RoboticsLabURJC
Programming Robot Intelligence

End-to-end Vision-based Autonomous Driving using Deep Learning

Author
Sergio Paniego Blanco
Supervisor
José María Cañas Plaza
Doctoral Program in Information and Communications Technologies
International Doctoral School
2024
@sergiopaniego
sergio.paniego@urjc.es
https://sergiopaniego.github.io/

E2E vs modular

- "End-to-end Autonomous Driving: Challenges and Frontiers"
- Advantages:
  - Simplicity in combining perception, prediction, and planning into a single model.
  - The whole system is optimized towards the ultimate task of planning/control.
  - More computationally efficient.
  - Data-driven optimization has the potential to improve the system by simply scaling training resources.
- Limitations/Challenges:
  - Handling different input modalities.
  - Visual abstraction.
  - World model learning.
  - Multi-task frameworks.
  - Inefficient experts and Policy distillation.
  - Lack of interpretability.
  - Causal confusion.
  - Lack of Robustness.
  - …

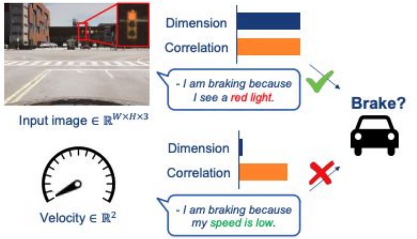


Fig. 7: **Causal Confusion.** The current action of a car is strongly correlated with low-dimensional spurious features such as the velocity or the car's past trajectory. End-to-End models may latch on to them leading to causal confusion.

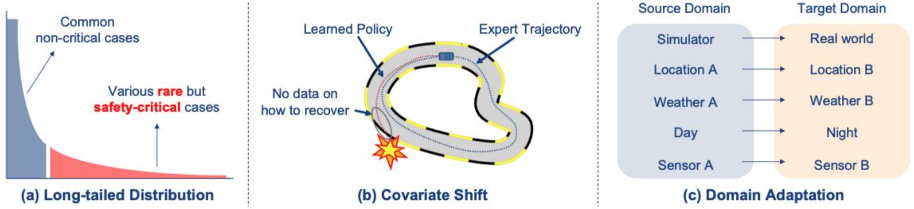


Fig. 8: **Challenges in robustness.** Three primary generalization issues arise in relation to dataset distribution discrepancies, namely long-tailed and normal cases, expert demonstration and test scenarios, and domain shift in locations, weather, *etc.*

Transferring to real world

- Domain adaptation.
- Increase sensor coverage.
- Incorporate real-world data into training.

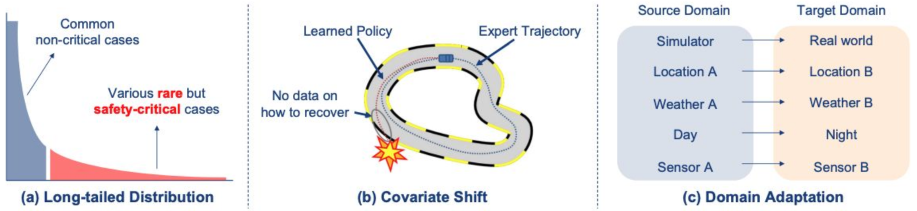


Fig. 8: **Challenges in robustness.** Three primary generalization issues arise in relation to dataset distribution discrepancies, namely long-tailed and normal cases, expert demonstration and test scenarios, and domain shift in locations, weather, *etc.*

Universidad
Rey Juan Carlos

RoboticsLabURJC
Programming Robot Intelligence

End-to-end Vision-based Autonomous Driving using Deep Learning

Author
Sergio Paniego Blanco
Supervisor
José María Cañas Plaza
Doctoral Program in Information and Communications Technologies
International Doctoral School
2024
@sergiopaniego
sergio.paniego@urjc.es
https://sergiopaniego.github.io/

Universidad
Rey Juan Carlos

RoboticsLabURJC
Programming Robot Intelligence