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/

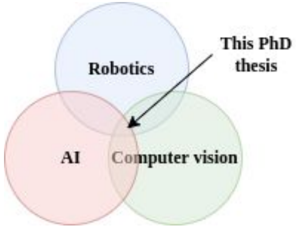
Outline

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

Introduction

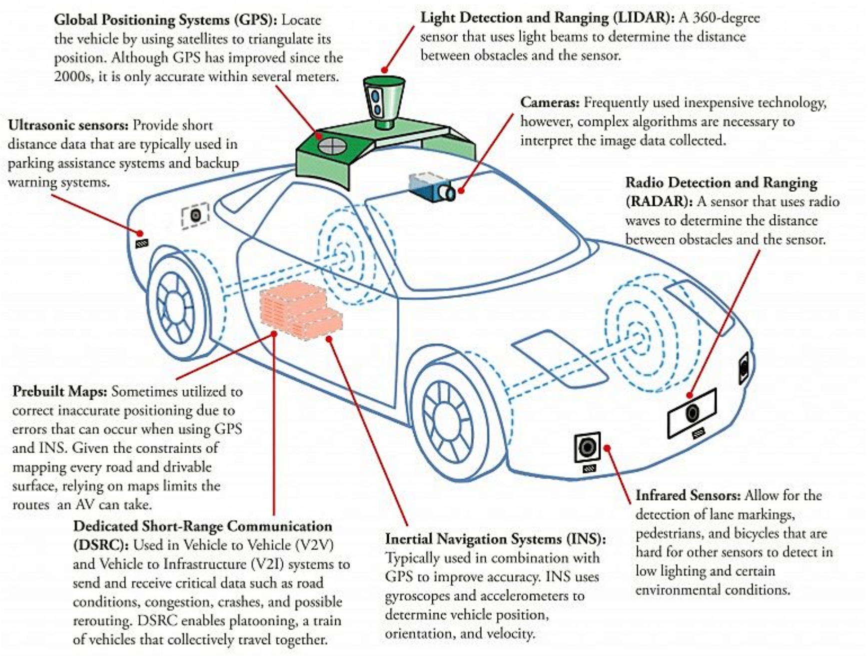
Introduction

- **Autonomous driving (AD):** vehicles that drive safely without human intervention.
- **Robotics + Artificial Intelligence (AI) + Computer Vision (CV).**
- 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.
  - …
- Some examples from AD companies already suggest an advancement in this direction.



Introduction to AD

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



Levels of autonomy

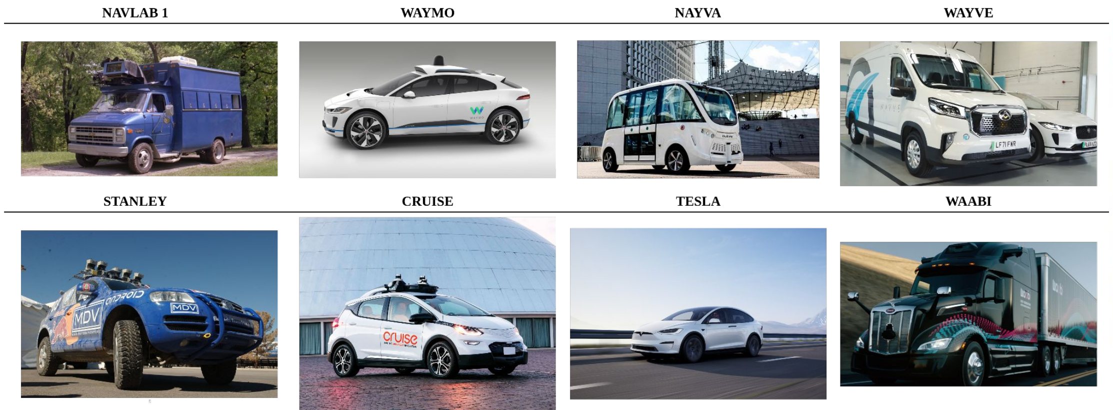
LEVEL 0	LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5
No automation	One automatic task at a time	Two automatic tasks at a time	The vehicle controls most driving tasks	The vehicle controls all driving tasks under certain conditions	The vehicle controls all driving tasks under all conditions

History

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

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 humans passengers).



Growing research community

- Incremental importance.
- Many workshops, tutorials in conferences.
- Competitions.

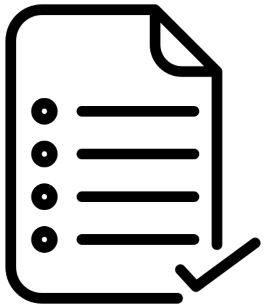
AWS DeepRacer

F1TENTH

DuckieTown

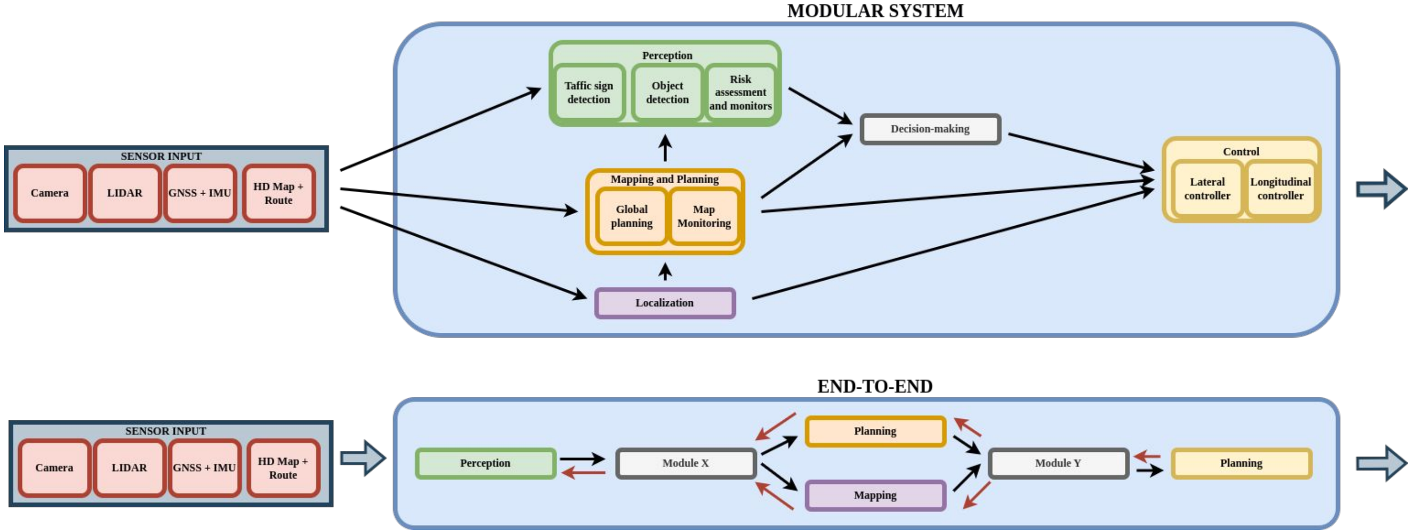
Assessment and metrics

- 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.
- 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.



Tasks in an AD system

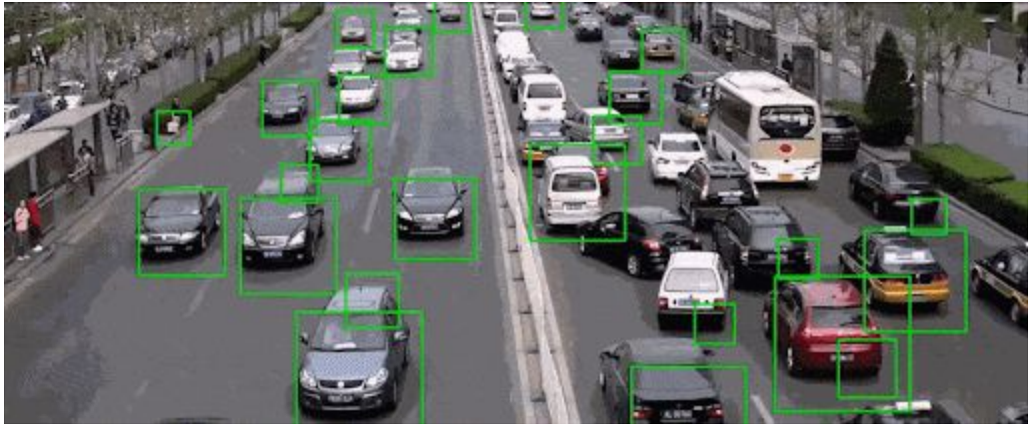
- A multitude of driving tasks are combined into the final AD system.
- Autonomous parking, lane following, intersection negotiation…

Optimization in AD solutions

- In robotics and AD, we need solutions that are fast and reliable.
- The performance not only depend on the quality of the decisions but also on their frequency.
- Some AD vehicles and robots are equipped with high-performance hardware but there are systems where this is not true.
- Solutions must be optimized while upholding stringent safety standards.

Traffic monitoring using CV

- Traffic monitoring plays a pivotal role in urban planning transportation managements, and public safety.



Research goals

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

1. A systematic study of the state-of-the-art in AD.
2. Validate CV in a driving setting generating a traffic monitoring tool.
3. Generate an object detection assessment software to validate solutions.
4. Generate an autonomous driving behaviors assessment software to conduct the experiments and generate quantitative data.
5. Empirical study of the improvements to the basic visual lane following with the addition of visual memory and kinematic input.
6. Empirical study of model optimization for the controllers of AD vehicles, generating optimized models that improve the system.
7. Generation of and E2E shallow model capable of driving in traffic based on visual input.

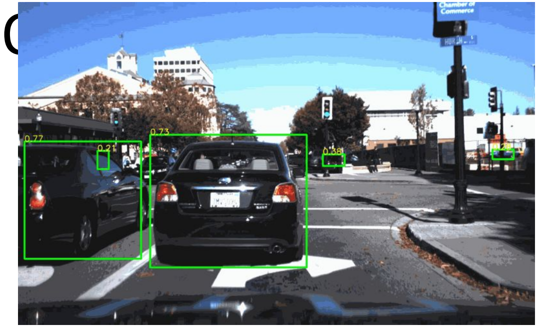
State-of-the-art

Traffic monitoring

- Traffic monitoring is a classic problem within computer vision.
- Before deep learning, traffic surveillance using video cameras was significantly limited but this changed with the appearance of this technology.

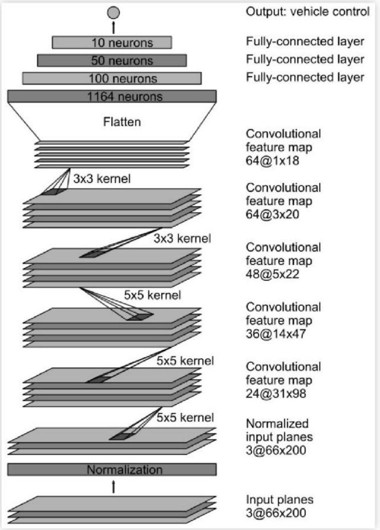
Deep learning object detection, datasets and assessment

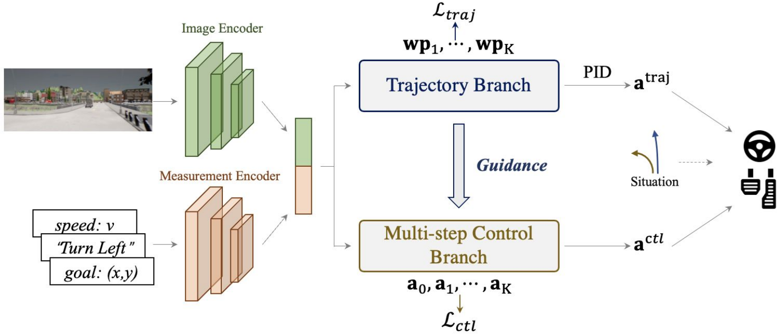
- Significant popularity lately. Notable advancements.
- Part of the perception in autonomous driving systems.
- Many datasets are available like COCO, ImageNet, Pascal VOC… and some others specialized.
- Each CV tasks includes a specific set of evaluation metrics.
- All the current state-of-the art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



Autonomous driving and imitation learning

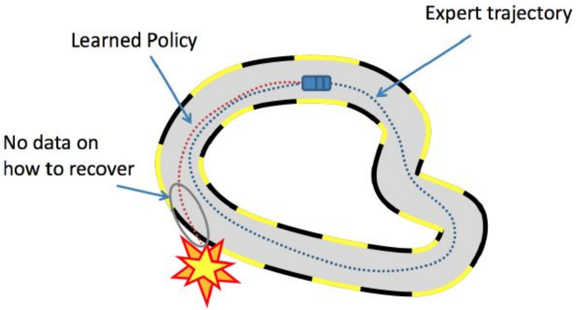
- AD combines artificial intelligence, robotics, and CV.
- Modular vs E2E approach.
- Vision-based E2E AD.
- PilotNet.
- TCP, ReasonNet, NET, Transfuser…

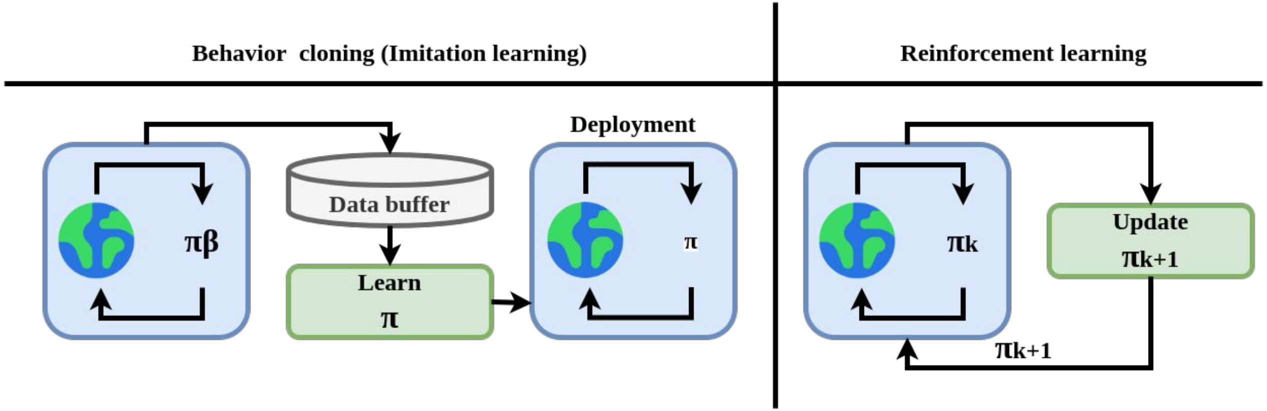




Imitation learning and RL for driving autonomously

- Imitation learning (behavior cloning): generating a driving policy learning from expert agent-gathered data.
- Reinforcement learning (RL).
- Combination of both.



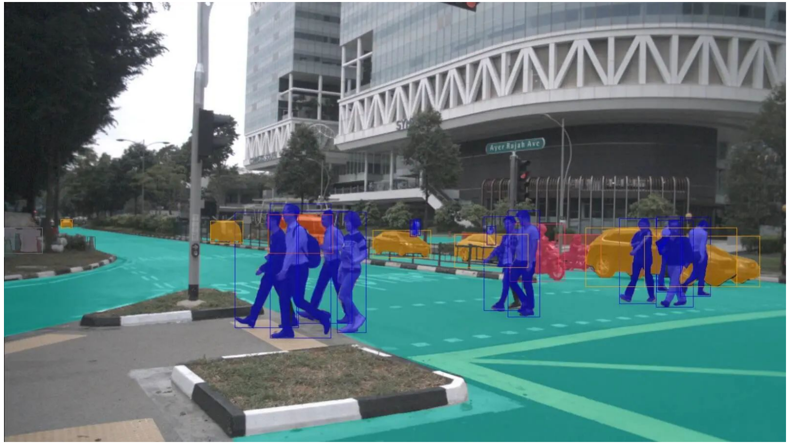


Simulation in AD and assessment. Datasets

- In the development and research of robotics systems, simulators are commonly used to generate solutions rather than directly developing in real robotics or vehicles.
- A diverse array of simulators for AD are available. SUMO, TORCS, Gazebo, CARLA, DeepDrive, Autoware, proprietary simulators, world generators…

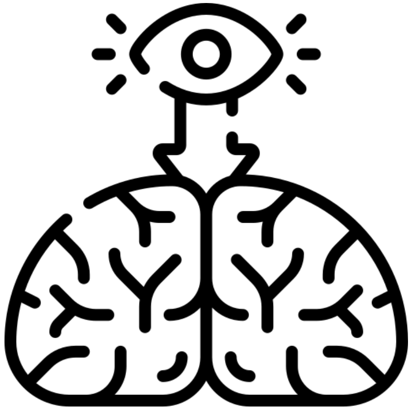
Simulation in AD and assessment. Datasets

- 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.



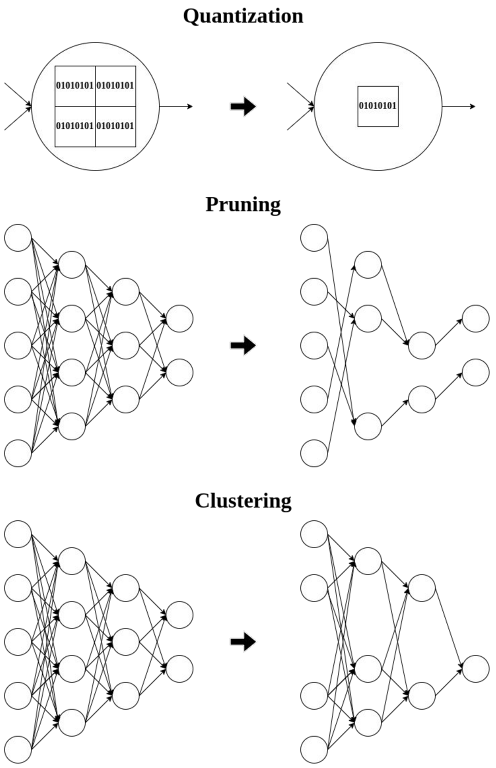
Memory-based approaches in E2E visual AD

- An intriguing avenue of research within the E2E approach involves integrating memory capabilities.
- LSTM modules, convLSTMS…
- Modified input.



Optimization of deep learning models for autonomous driving

- The optimization of deep learning models is key, as previously discussed.
- There are an extensive number of optimization techniques:
  - Quantization.
  - Pruning.
  - Fine-tuning (re-training) with optimization aware techniques.
  - Clustering.
- PyTorch or Tensorflow already include support.
- TensorRT for Nvidia GPUs.



TrafficSensor

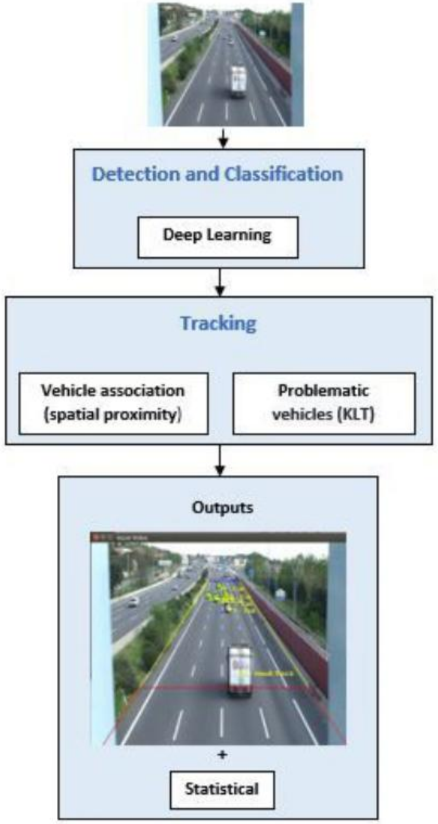
Monitoring and assessing traffic with deep learning

- 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).

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

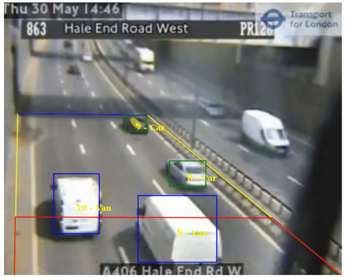
Monitoring and assessing traffic with deep learning

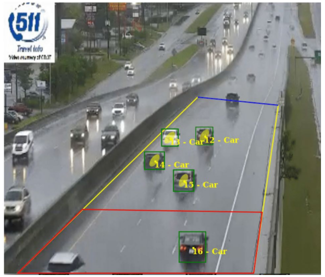
- New dataset created (7 vehicle types).
- Two modules:
  - Vehicle detection and classification.
  - Vehicle tracking.
- 1st module: YOLOv4.
- 2nd module: simple spatial association algorithm and more sophisticated KLT tracker to follow the vehicles.
- Several experiments validated the system.



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.

DetectionMetrics

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

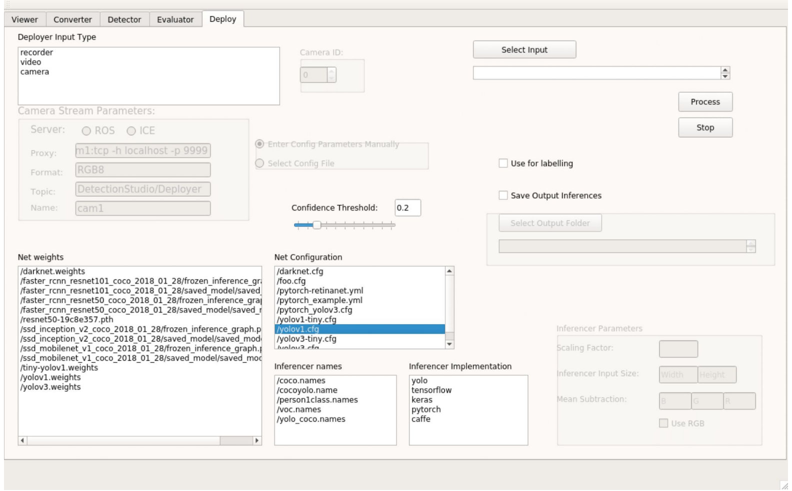
- Objective performance metrics.
- Systematic evaluation of different DL models for object detection on large datasets.
- Broad range of applicability: traffic monitoring, autonomous driving perception…
- Suite of tools with unique features to facilitate the objective comparison of different DL models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

Detection Metrics tool kit

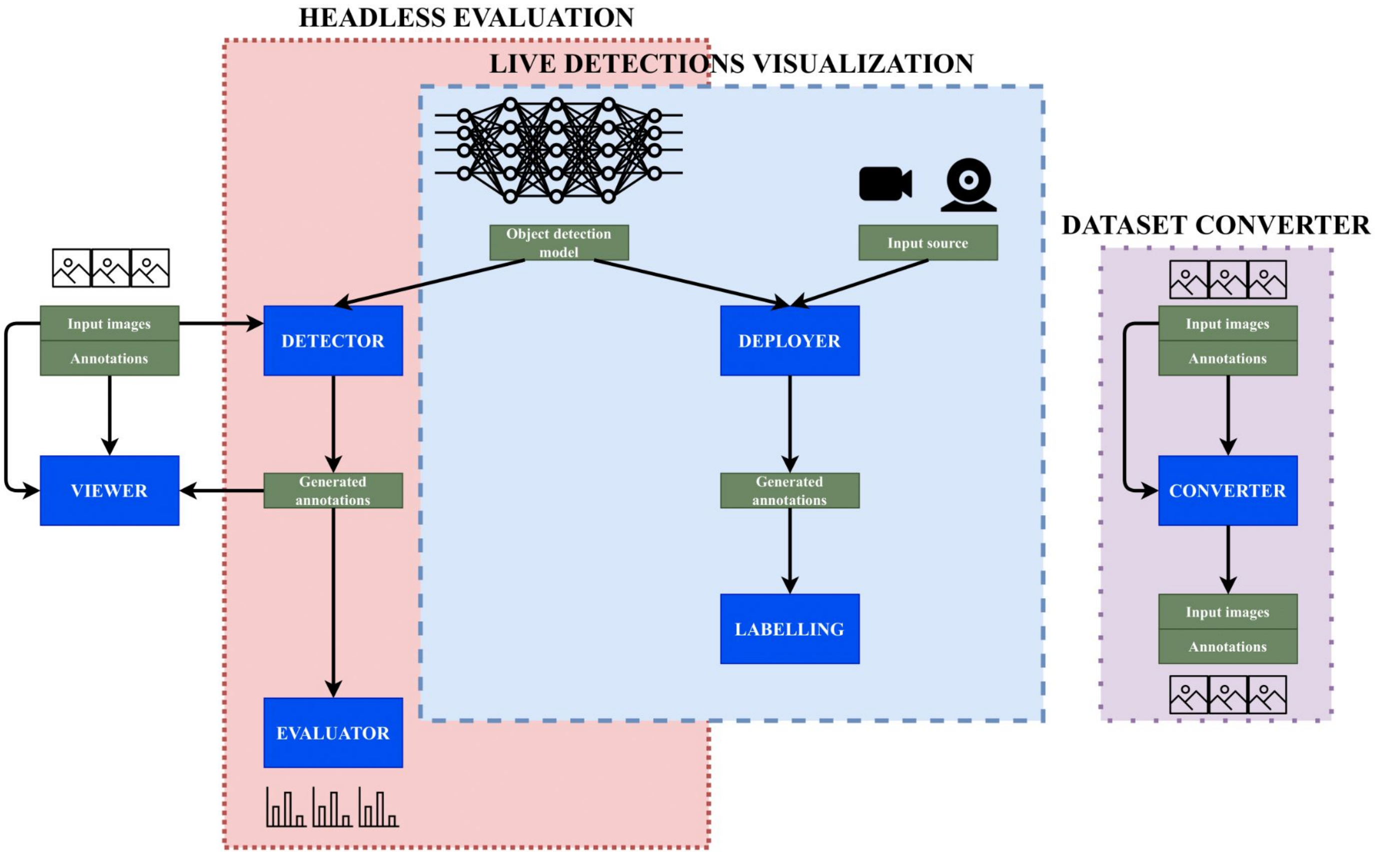
- DL frameworks support.
- Object detection datasets support.
- It can be used as a ROS Node.
- Two workflows:
  - Headless.
  - Graphical.
- Main tools:
  - Detection generation.
  - Evaluation of detections with objective metrics.
  - Live detection visualization.
  - Dataset converter.

IMAGENET

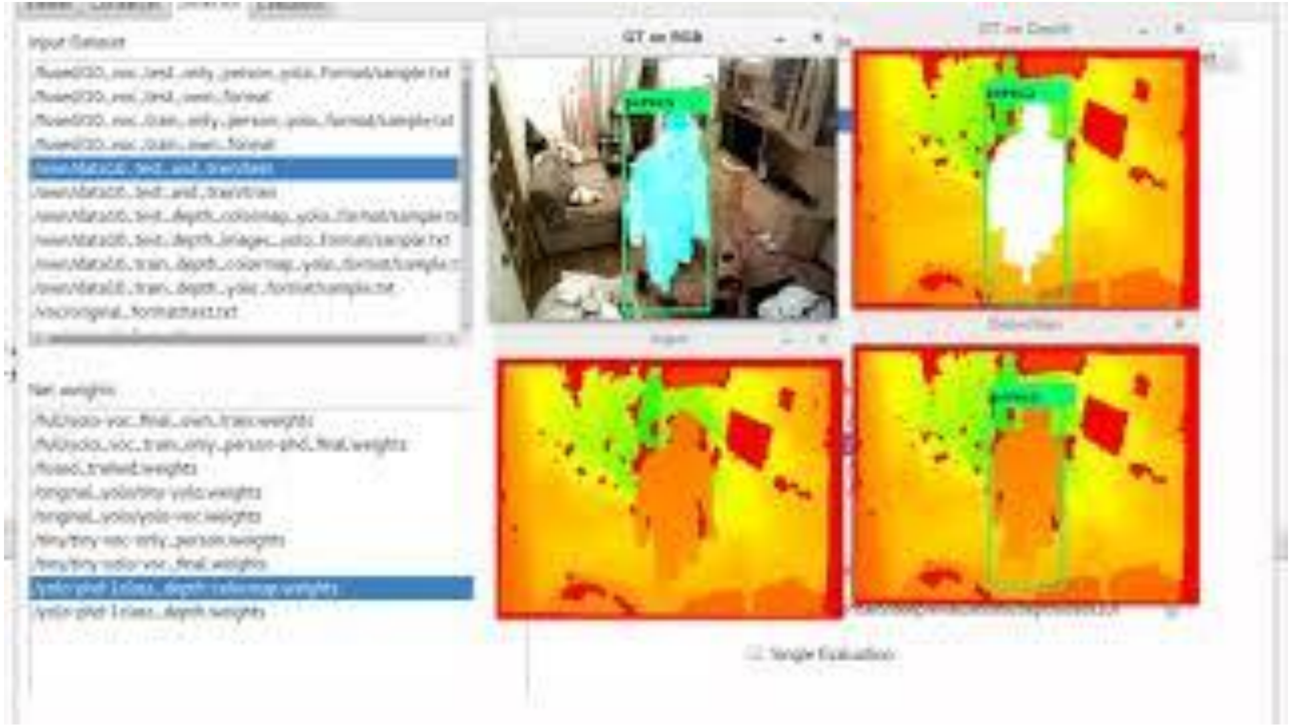
PASCAL2
Pattern Analysis, Statistical Modelling and Computational Learning



Detection Metrics tool kit

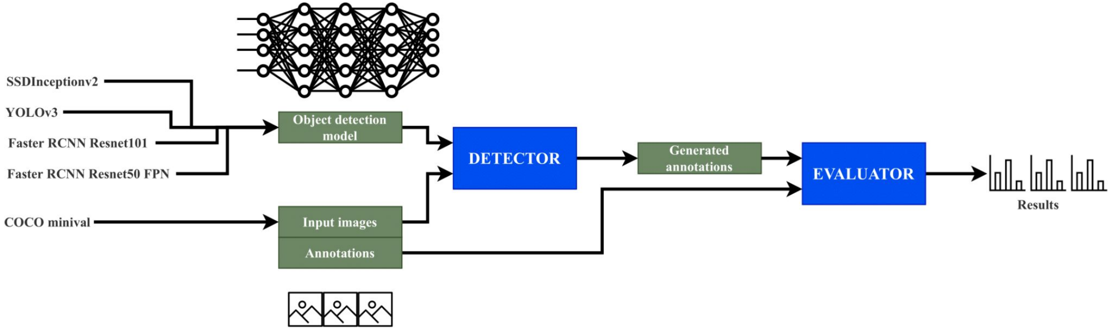


Detection Metrics tool kit



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.

Behavior Metrics

Assessing autonomous driving behaviors fine-grained metrics

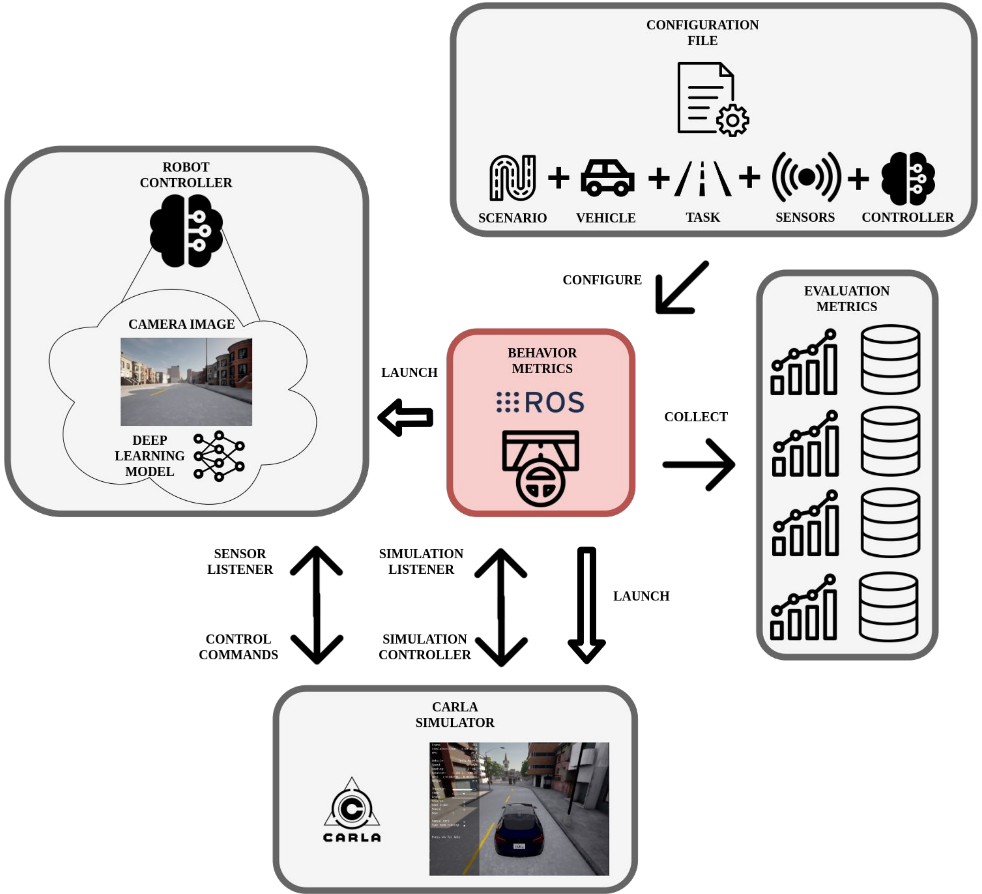
- **Behavior Metrics** is an open-source software tool developed to help research in the autonomous driving field.
- The development and validation of autonomous driving solutions require testing broadly in simulation (online).
- Quantitative and qualitative assessment and comparison of solutions for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

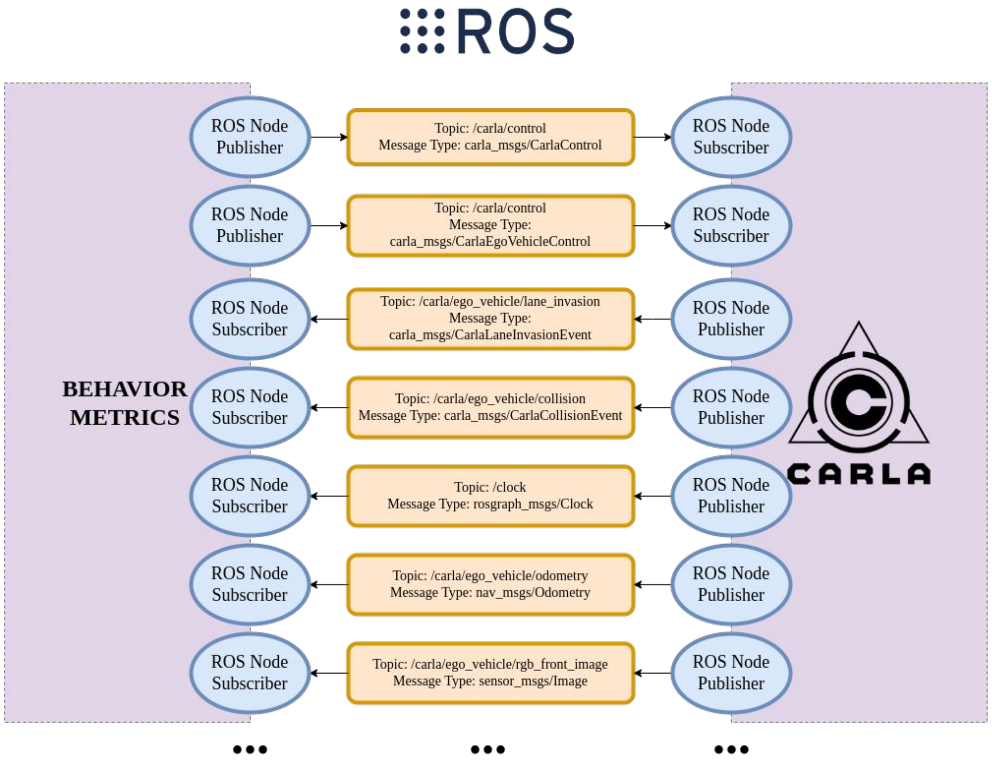
Assessing autonomous driving behaviors fine-grained metrics

- 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.

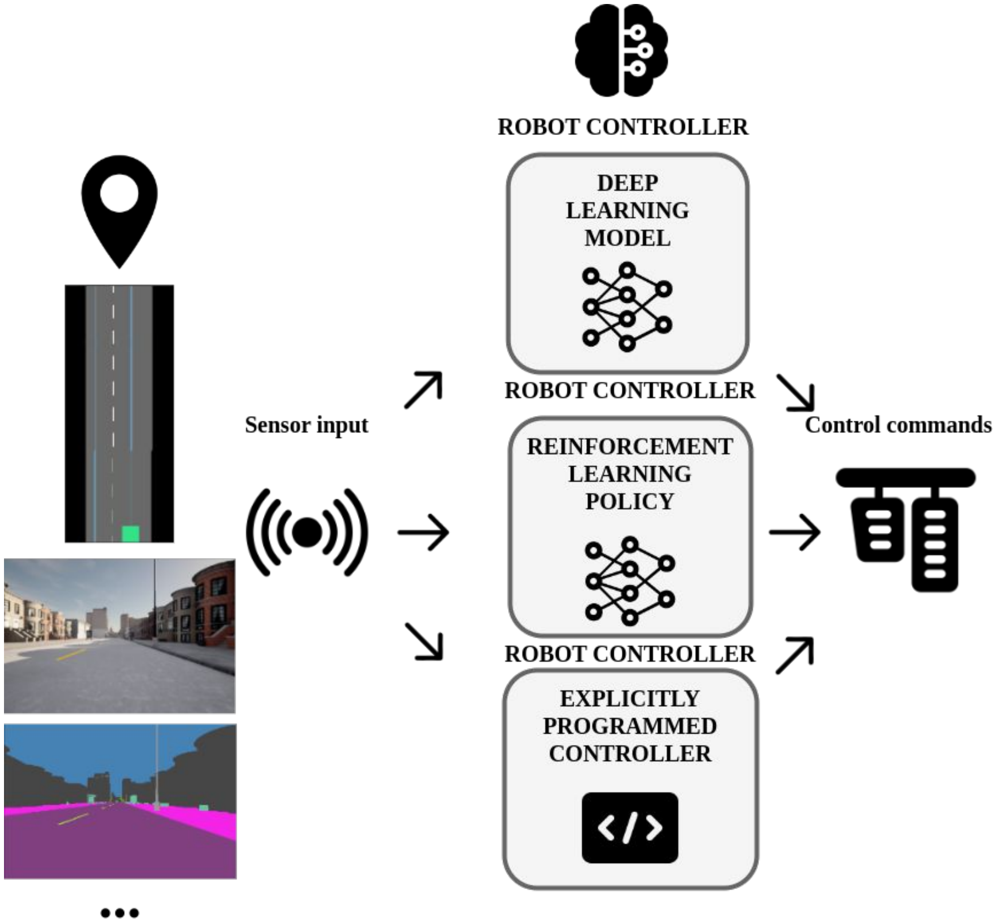
Assessing autonomous driving behaviors fine-grained metrics



Connection between BM and CARLA



Robot controller

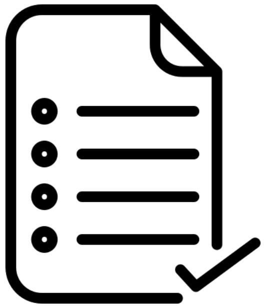


Supported tasks

- 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.
    - …

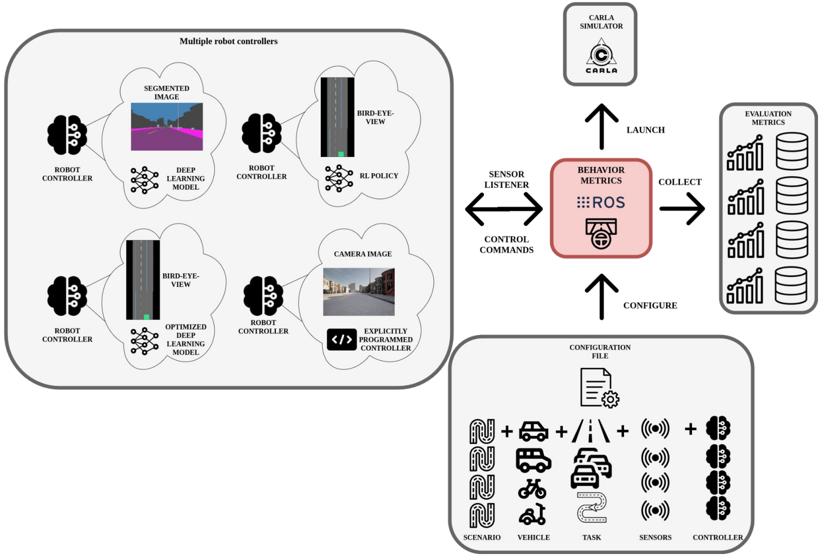


GUI evaluation mode

JdeRobot

Behavior Metrics GUI mode

Headless evaluation mode



Behavior Metrics headless mode

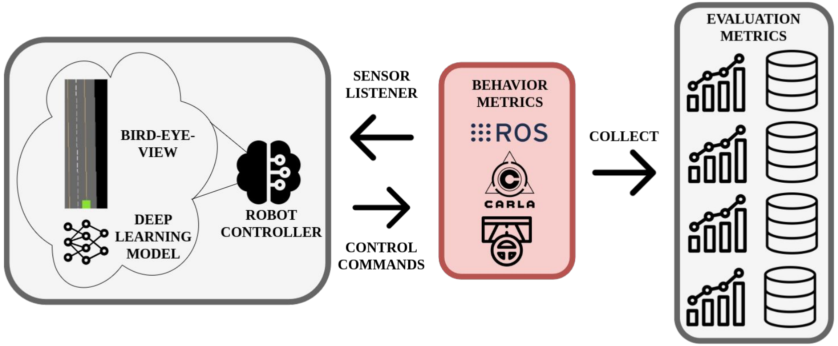
Impact and conclusions

- Used as an assessment tool in this thesis.
- There is a need for specialized assessment tool in autonomous driving for easy extraction of conclusions from experiments using fine-grained metrics.
- Open-source.

Adding memory

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

- **Explored and compared various approaches to enhance the capabilities of an end-to-end system for autonomous driving 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

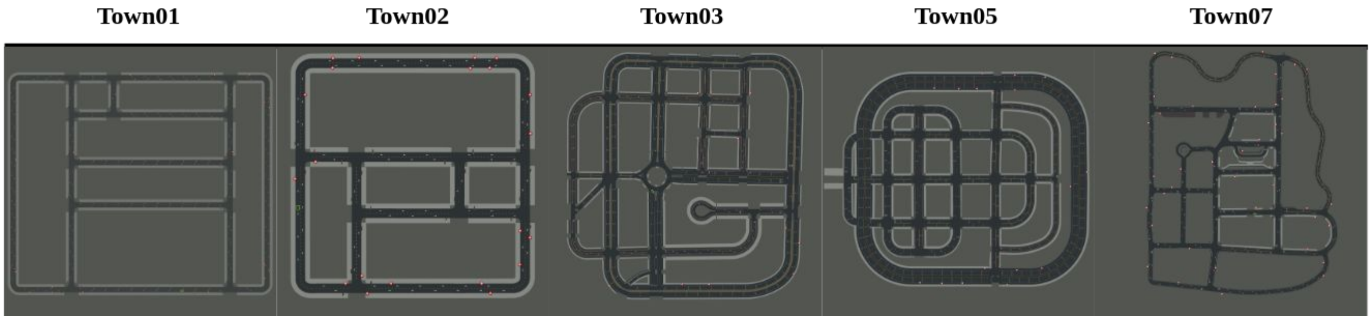
- Lane-following application using different urban scenario layouts and visual bird-eye-view input.
- Memory addition:
  - Architectural modifications.
  - Different types of sensory input.
- Incorporating visual memory capabilities and kinematic input data: system more robust and able to handle a wider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

Introduction

- We introduce and compare several DL architectural modifications that incorporate memory models and different sensory inputs to explore the potential benefits of different memory approaches in solving the problem of end-to-end AD through imitation learning.
- We also explore the inclusion of kinematic data as input.
- Hypothesis: adding visual memory capabilities and kinematic input data improves the quality of the generated robot control behavior in terms of the system's robustness for never-seen situations and speed self-regulation.

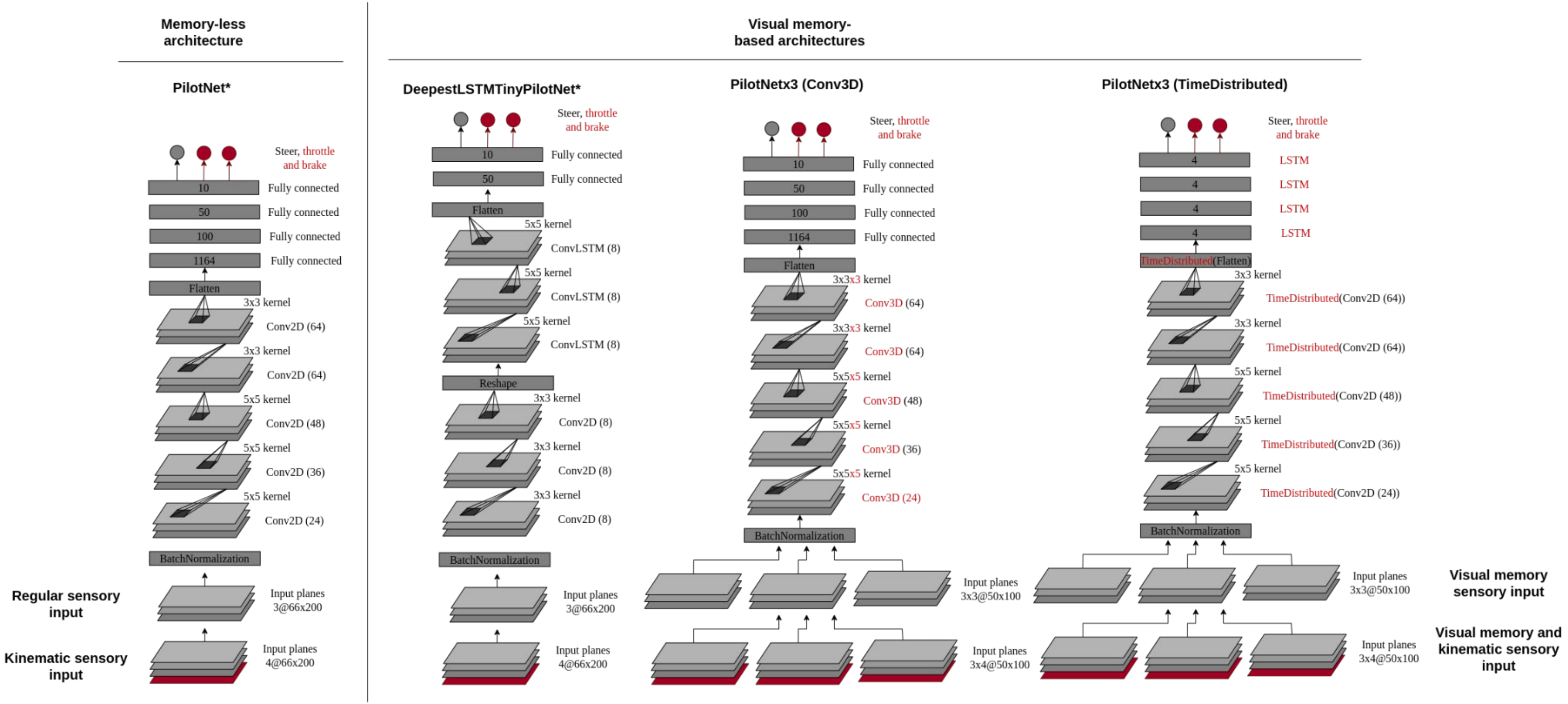
Kinematic-infused and visual memory E2E control based on imitation learning

- 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.



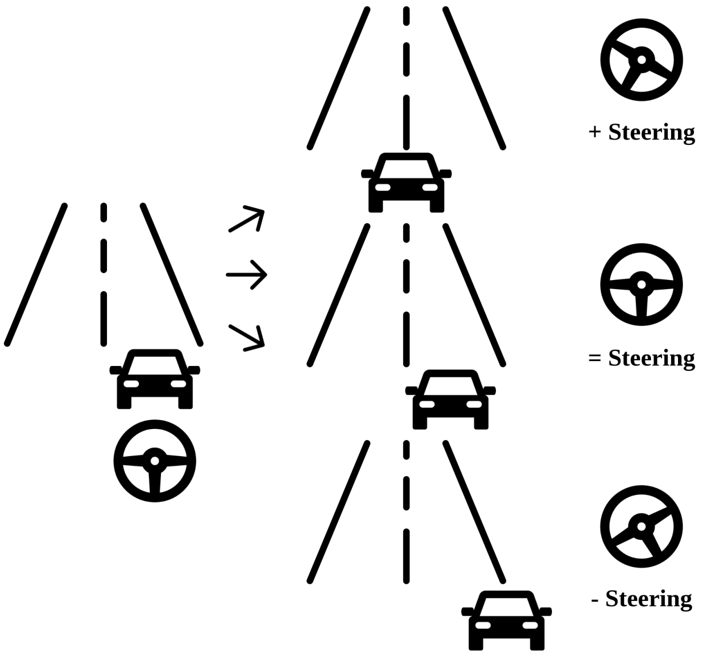
Kinematic input and memory-based architectures

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 is key.



Experiment 1

- Comparison of models using common ML metrics:

Model	Visual memory	Kinematic input	MAE test	MSE test
PilotNet*	✘	✘	0.0507	0.0177
PilotNet*	✘	✔	0.0332	0.0086
DeepestLSTMTinyPilotNet*	✔	✘	0.0662	0.0196
DeepestLSTMTinyPilotNet*	✔	✔	0.0456	0.0094
PilotNetx3* (Conv3D)	✔	✘	0.0295	0.0082
PilotNetx3* (Conv3D)	✔	✔	0.0074	0.0079
PilotNetx3* (TimeDistributed)	✔	✘	0.0289	0.0077
PilotNetx3* (TimeDistributed)	✔	✔	0.0069	0.0086

Table 6.1: MAE and MSE metrics comparison for each trained model using test data from the dataset. Four different architectures are tested with different input data considerations: bird-eye view (BEV) and velocity sensory data. ✔: supported. ✘: unsupported.

Experiment 2

- Behavior in test scenario with top speed regulation

Map	**Town02**							
Model	**Pilotnet***		**DeepestLSTMTinyPilotNet***		**Pilotnetx3* (Conv3D)**		**Pilotnetx3* (TimeDistributed)**	
Visual memory	✘	✘	✔	✔	✔	✔	✔	✔
Kinematic input	✘	✔	✘	✔	✘	✔	✘	✔
Effective completed distance (m)	820.6	868.9	830.6	846.6	902.6	875.2	889.0	852.3
Position deviation mean per km (m/km)	0.25	0.26	0.22	0.33	0.24	0.25	0.29	0.29
Controller iterations frequency (Hz)	18.32	18.40	18.25	18.05	17.10	17.05	17.65	17.51
Vehicle jerk in control commands per kilometer	0.31	0.19	0.16	0.15	0.18	0.12	0.19	0.12
Vehicle jerk in velocity per kilometer	0.34	0.33	0.30	0.31	0.33	0.49	0.34	0.51
Average speed (km/h)	24.71	26.78	25.01	26.41	27.51	26.77	27.15	26.21
Max speed (km/h)	31.35	30.08	31.30	29.92	31.57	31.25	31.50	30.98
Experiments with collisions	0/5	0/5	0/5	0/5	0/5	0/5	0/5	0/5
Collisions per km	0	0	0	0	0	0	0	0
Lane invasions per km	0	0.46	0	3.08	0	0.46	0	0
Successful experiments	**5/5**	**5/5**	**5/5**	**5/5**	**5/5**	**5/5**	**5/5**	**5/5**

Table 6.2: Comparison of models (columns) in different test environments considering some measured metrics (rows) provided by Behavior Metrics. Values in **bold** highlight the most interesting results. ✔: supported. ✘: unsupported.

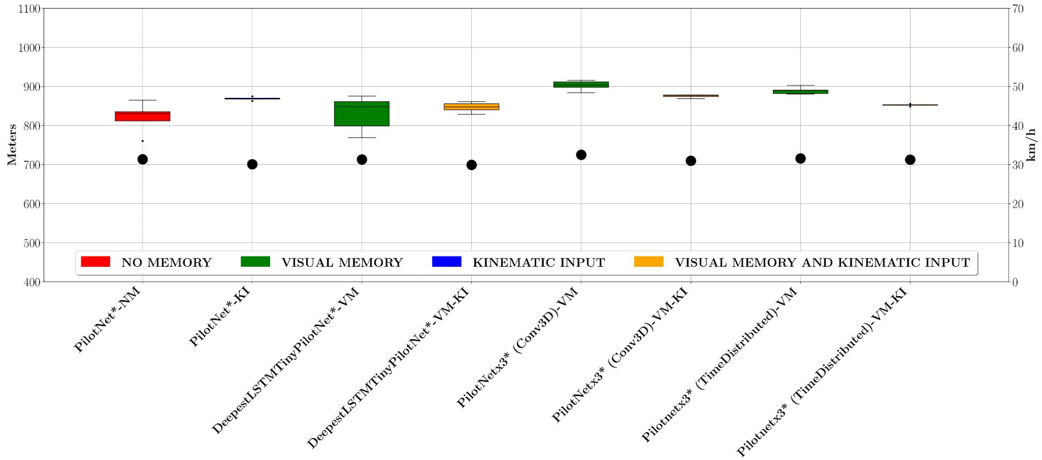
Experiment 3

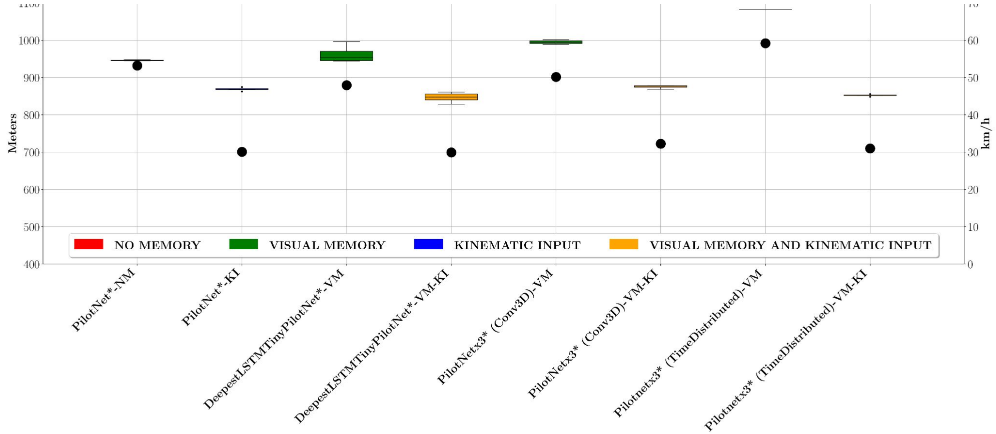
- Behavior in test scenario without top speed regulation

Map	Town02							
Model	Pilotnet*		DeepestLSTMTinyPilotNet*		Pilotnetx3* (Conv3D)		Pilotnetx3* (TimeDistributed)	
Visual memory	✘	✘	✔	✔	✔	✔	✔	✔
Kinematic input	✘	✔	✘	✔	✘	✔	✘	✔
Effective completed distance (m)	**946.3**	868.9	**962.0**	846.6	**994.8**	875.2	**1083.0**	852.3
Position deviation mean per km (m/km)	**0.23**	0.26	**0.21**	0.33	**0.28**	0.25	**0.28**	0.29
Controller iterations frequency (Hz)	**17.34**	18.40	**16.11**	18.05	**17.09**	17.05	**17.65**	17.51
Vehicle jerk in control commands per kilometer	**0.28**	0.19	**0.15**	0.15	**0.18**	0.13	**0.17**	0.12
Vehicle jerk in velocity per kilometer	**0.26**	0.33	**0.22**	0.31	**0.25**	0.49	**0.21**	0.51
Average speed (km/h)	**30.37**	26.78	**30.13**	26.41	**31.72**	26.77	**38.82**	26.21
Max speed (km/h)	**53.20**	30.08	**47.93**	29.92	**50.15**	31.25	**59.18**	30.98
Experiments with collisions	**5/5**	0/5	**1/5**	0/5	**3/5**	0/5	**4/5**	0/5
Collisions per km	**2.09**	0.0	**0.32**	0.0	**0.95**	0.0	**1.45**	0.0
Lane invasions per km	**2.24**	0.46	**0.71**	3.08	**0.87**	0.46	**1.05**	0.0
Successful experiments	**0/5**	**5/5**	**0/5**	**5/5**	**0/5**	**5/5**	**0/5**	**5/5**

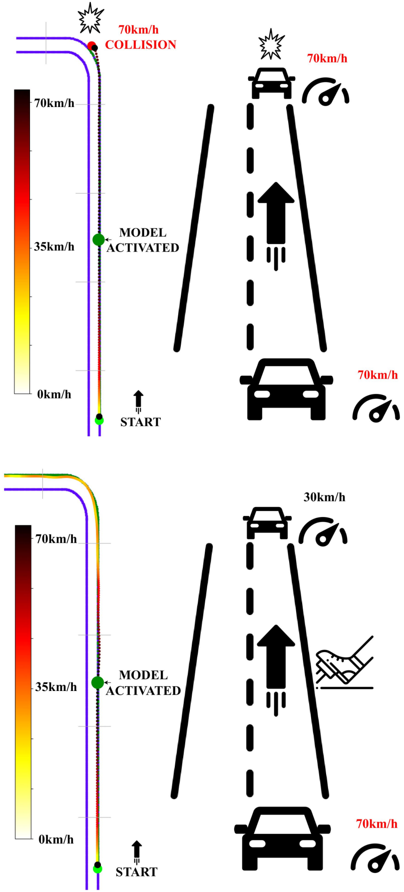
Table 6.3: Comparison of models in different test environments without top speed limit considering metrics from Behavior Metrics. **Bold** values (excluding *Successful experiments*) indicate changes in results from previous experiment results. Values in **red bold** and **bold** for *Successful experiments* highlight the most interesting results. ✔: supported. ✘: unsupported.

Top speed regulation vs no regulation





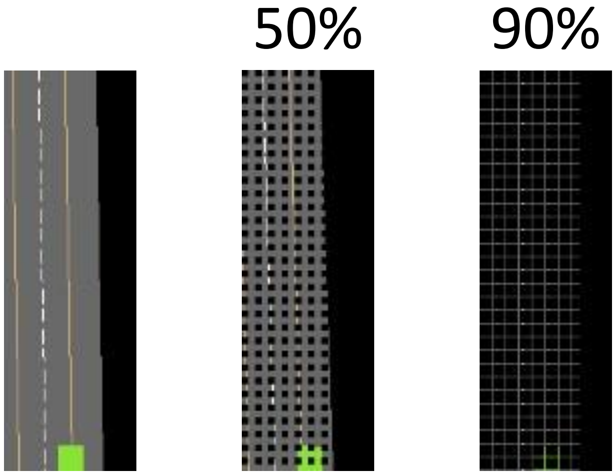
Taking control of a fast-moving car



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



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 comparison

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 data to the models enhances the quality of the final control behavior for following the lane.
- Adding kinematic data, the system controls the speed better.
- Combining it with visual m 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.

Optimizing DL models

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

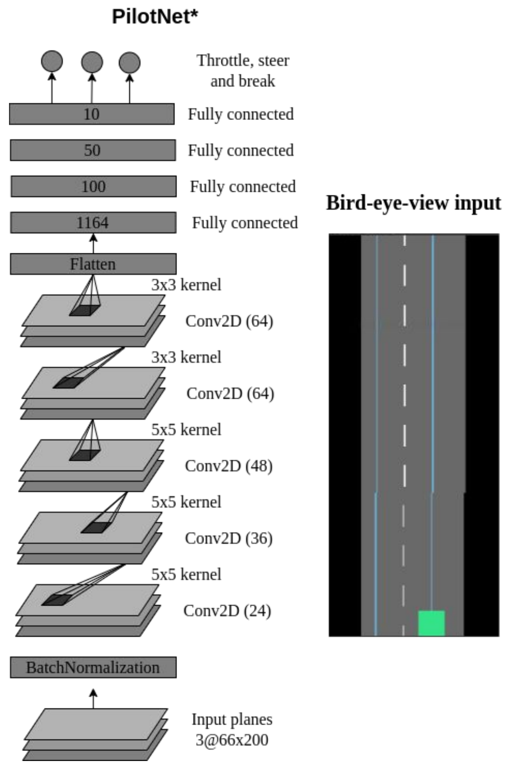
S. Paniego, N. Paliwal, and J. Cañas, "Model optimization in deep learning based robot control for autonomous driving," IEEE Robotics and Automation Letters and IEEE International Conference on Robotics and Automation (ICRA), vol. 9, no. 1, pp. 715–722, 2024. doi: 10.1109/LRA.2023.3336244. [Online].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

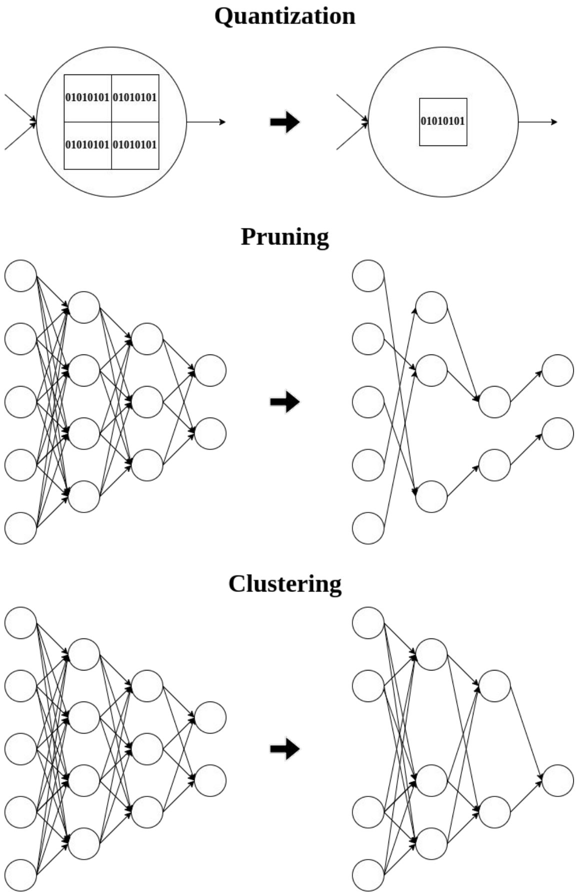
Optimizing E2E IL models for lane-follow robot control

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



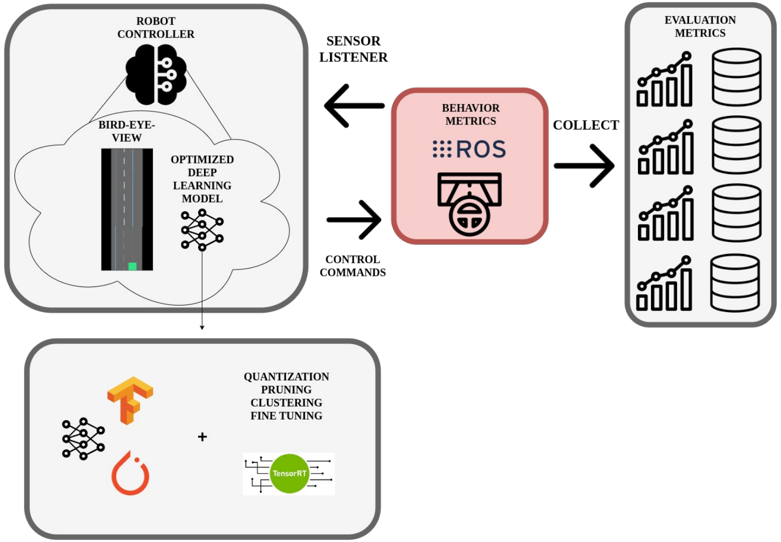
Optimizing E2E IL models for lane-follow robot control

- 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.

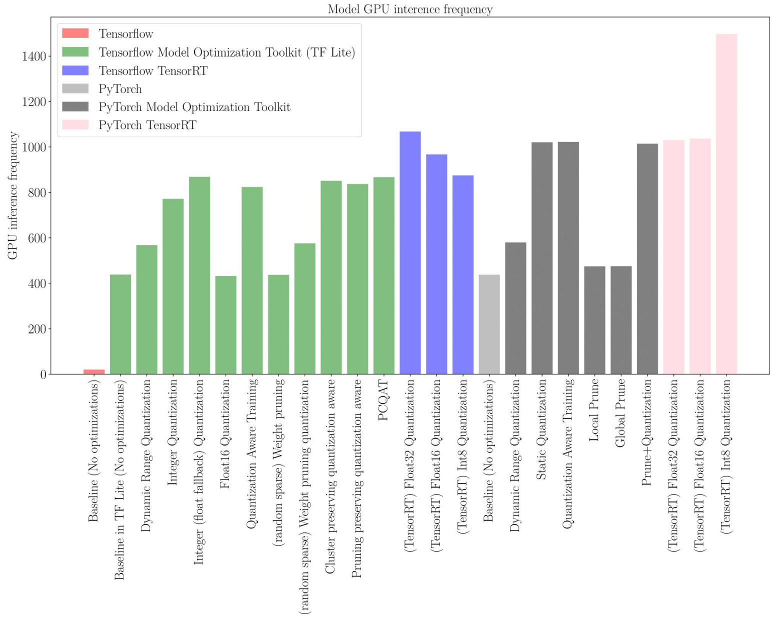


Experiment 1: Offline evaluation

- 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).

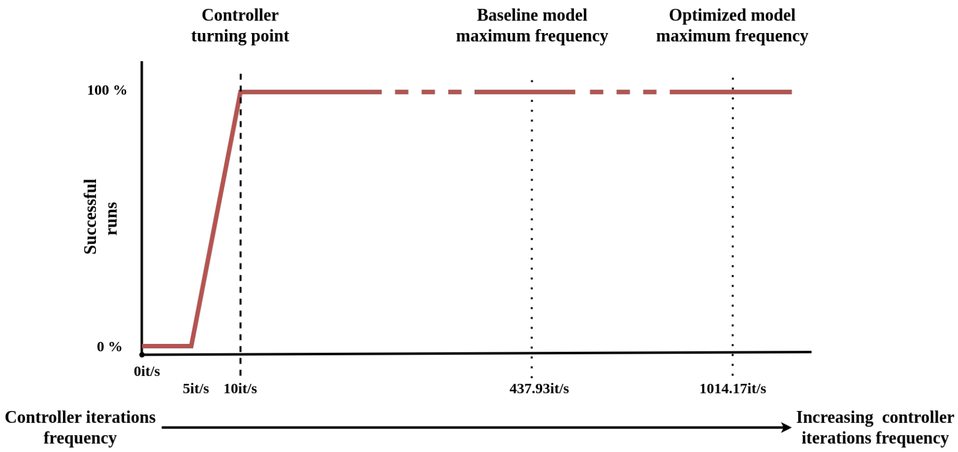
Experiment 2: Online evaluation

- 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.

Driving in traffic

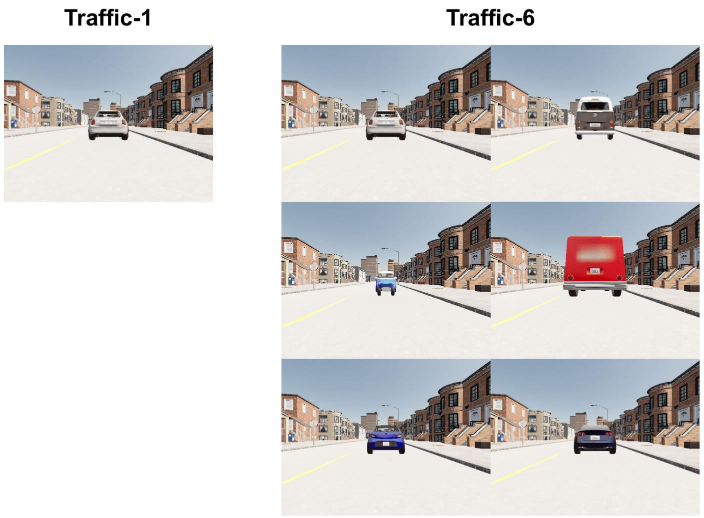
End-to-end vision-based autonomous driving in traffic

- **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.

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

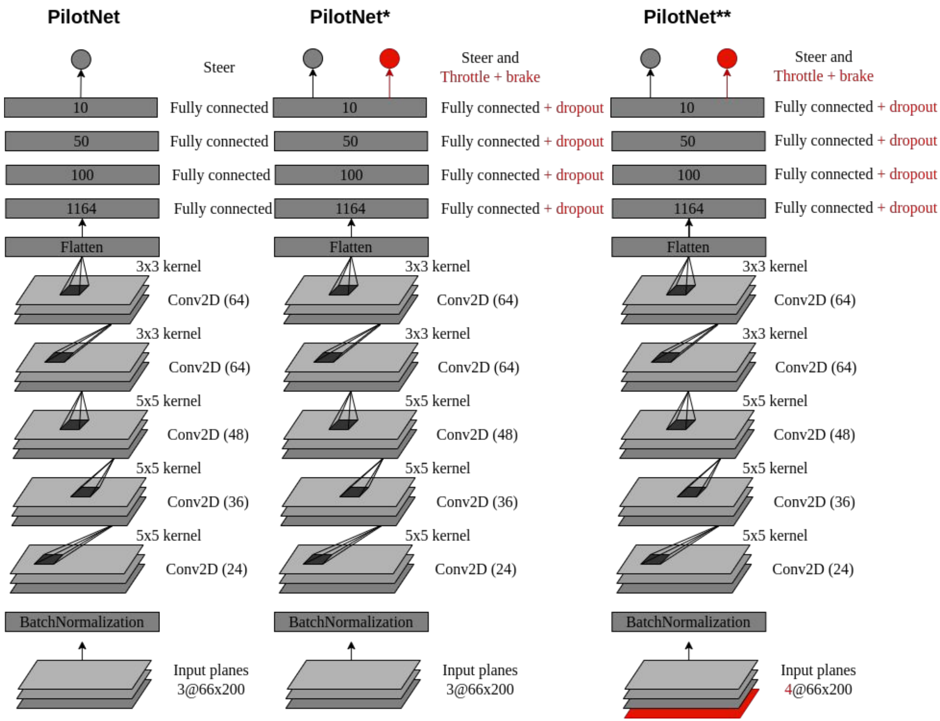
Imitation learning for driving in traffic

- 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.



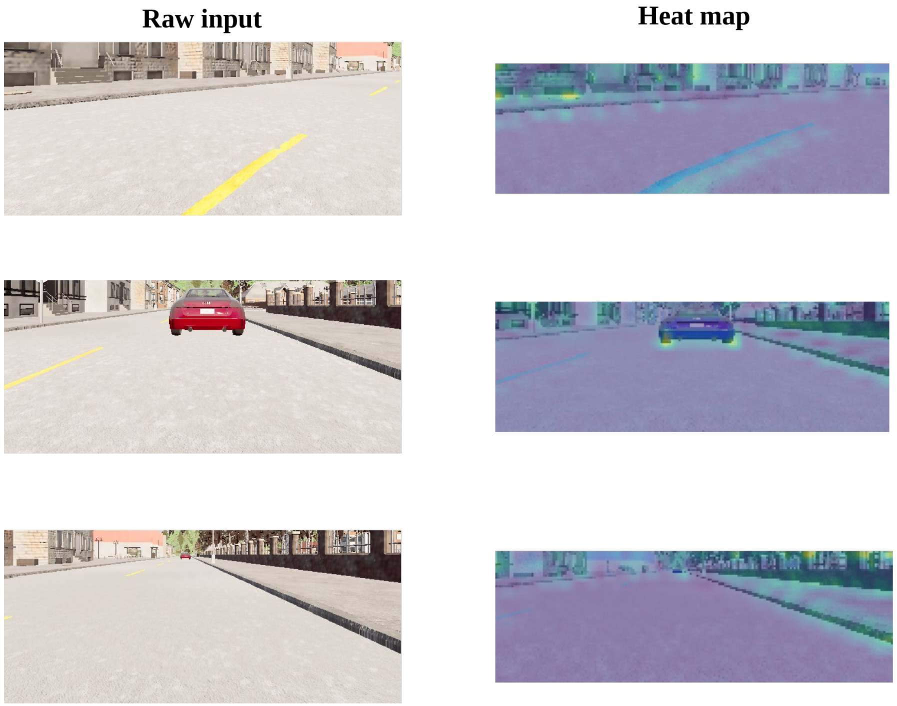
Imitation learning for driving in traffic

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



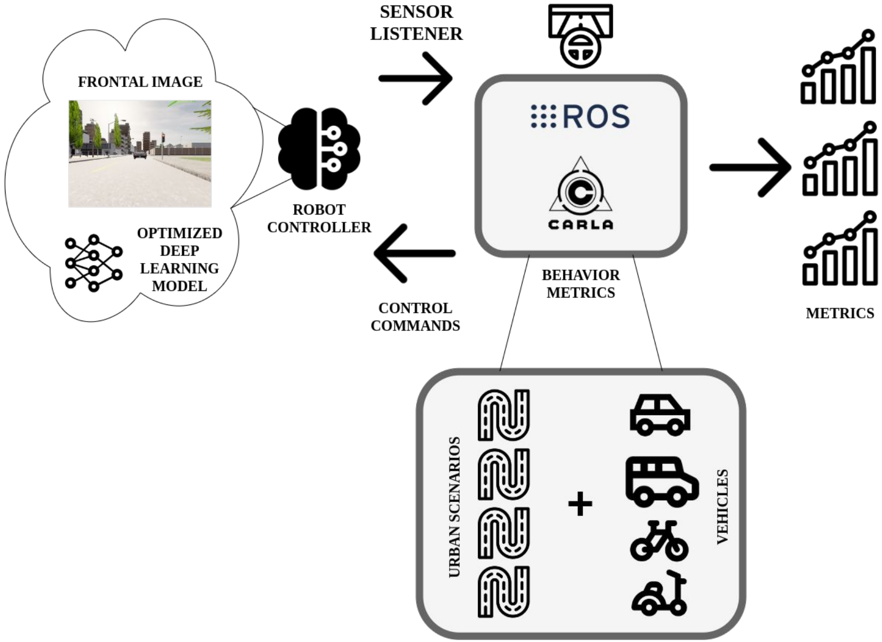
End-to-end vision-based autonomous driving in traffic

- 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 application area widely.



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

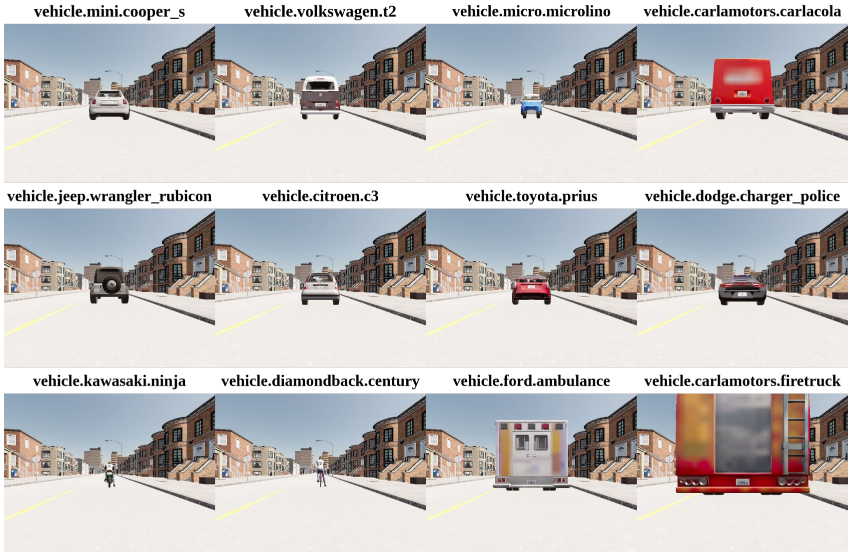


Table 8.5: Success rate metric for each of the 12 vehicles.

	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.

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.

1. Comprehensive literature review.
2. Solution for vehicle monitoring using deep learning.
3. Massive and unattended automatic assessment of DL visual object detection models solution.
4. Open source tool for the online assessment of AD systems.
5. Beyond achieving the basic follow lane, we explored the implications of the addition of visual memory and kinematic data to some apparently simple DL architectures.
6. Study the possibilities that optimization of DL add to the performance of the model and how they affect the final behavior.
7. Proposal for safe driving in traffic scenarios following an E2E vision-based approach with imitation learning and deep learning.

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

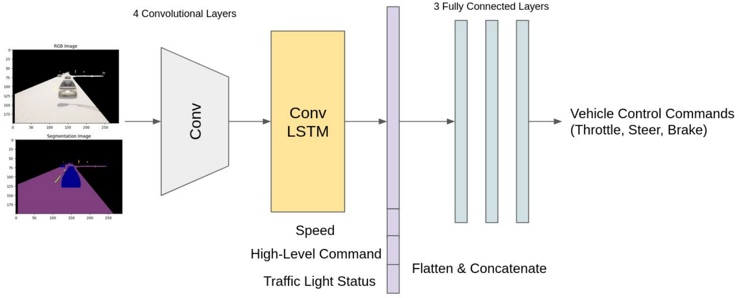
Research contributions

- **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

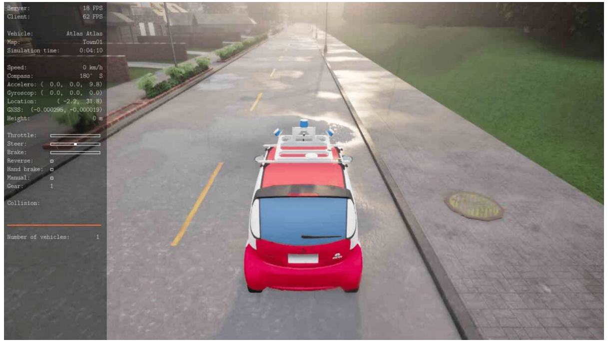
- Point-to-point E2E navigation using input commands.
- GSoC project.

Google
Summer of Code



Future (current) work 2

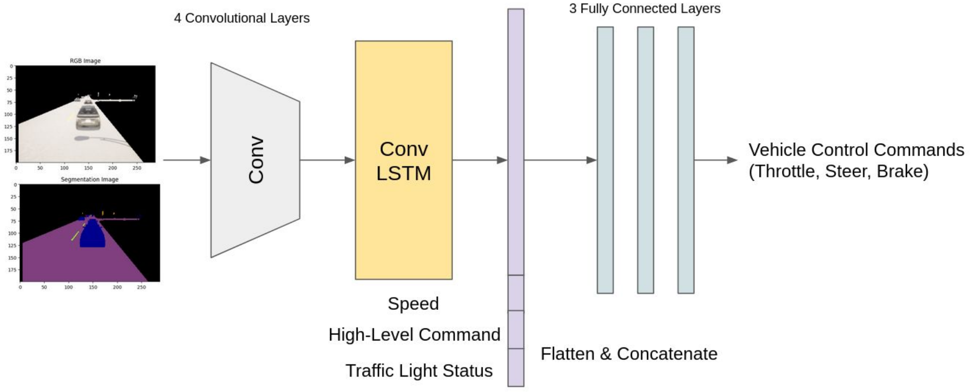
- Transferring current E2E solutions to a real-world vehicle.



Future (current) work 3

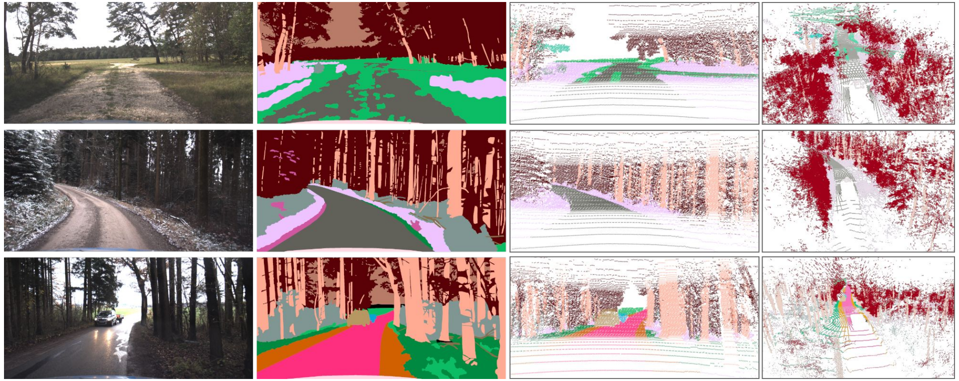
- E2E autonomous driving modulated with text-based instructions (LLMs).
- GSoC project.

Google



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.

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

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

Curso de Inteligencia Artificial

Sergio Paniego Blanco
@sergiopaniego
sergio.paniego@urjc.es
https://sergiopaniego.github.io/
https://roboticslaburjc.github.io/
Escuela de Ingeniería de Fuenlabrada

Teoría de la Señal y las Comunicaciones y
Sistemas Telemáticos y Computación

Outline

- Introducción a Python.
-

Web

- https://roboticslaburjc.github.io/

Universidad
Rey Juan Carlos

RoboticsLabURJC
Programming Robot Intelligence