Analytical Study on Region of Interest and Dataset Size of Vision-based End-to-End Lateral Control for Off-road Autonomy

Feeza Khan Khanzada[1], Bada Kwon[2], Woojin Jeong[3], Young Seek Cho[4], and Jaerock Kwon[1]

***Abstract*— Off-road autonomy is a challenging topic for mobile robots since the majority of navigation algorithms were developed for indoor, structured, or even surface outdoor environments. To address this problem, vision-based and Behavior Cloning (BC) style navigation with Deep Neural Networks (DNN) approaches have been proposed. Yet, it has not been clear which area of the input vision data must be focused on and how big the training dataset should be. In this study, we analyzed how variations in the ROI of the input image affect the controller's performance in terms of precision, completion time, and autonomy in off-road navigation. Our findings indicate that the selection of ROI significantly impacts the DNN controller's ability for off-road autonomous navigation. Specifically, we observe that full-sized input images tend to deteriorate the performance in precision driving tasks, capturing unnecessary details for maneuvering. Conversely, utilizing a cropped ROI, mainly focusing on the upper region of the bottom half of the image, can optimize completion time-related objectives. Furthermore, using a bigger dataset improved autonomy with a selection of the center area of ROI. These insights offer valuable considerations for designing a DNN-based BC controller tailored to specific navigation requirements, balancing performance, and real-world applicability.**

I. Introduction

Recent advancements in robotics and AI have significantly transformed various sectors, playing a vital role in improving human lives by handling rigorous or hazardous tasks. The agricultural sector has also witnessed substantial benefits from robotics technology. Robots contribute to enhancing food production for the growing population by undertaking tasks like precision planting, crop monitoring, weed detection, and autonomous harvesting. However, navigating in the agricultural environment poses unique challenges for mobile robots that have been developed for indoor environments due to the dense and uneven nature of fields, as well as varying lighting and weather conditions.

In response to these challenges, researchers have been exploring advanced methods like Deep Neural Network-based Behavior Cloning (DNN-based BC) [1][2] for autonomous navigation in agricultural landscapes [3][4][5][6][7]. Through comprehensive comparative analyses, we investigated how variations in Region of Interest (ROI) in the input image and dataset size impact the training of DNN-based BC algorithms for navigation in outdoor agricultural environments. By assessing different ROI and dataset variations, we aimed to provide insights into the selection of ROI and the effectiveness of dataset size in enhancing the control and navigation capabilities of robots in agricultural settings.

Conducting experiments in a simulated orchard farm environment, we measured the performance of DNN-based BC controllers. The metrics in this work are derived from [8], including driving deviation, completion time, success rate, and human intervention. Our results indicate that selecting the ROI and optimizing dataset size significantly influence the autonomous navigation performance of DNN-based BC algorithms in off-road agricultural environments, showing promising capabilities for enhancing robotic navigation in such challenging landscapes.

II. Related Work

Neural Network (NN)-based BC was proposed and validated by Autonomous Land Vehicle in a Neural Network (ALVINN) [9], which used a fully connected and shallow neural network. The input nodes consist of a 30×32 camera image and an 8×32 2D LiDAR scan image. One hidden layer with 29 nodes was used. The output unit has 45 nodes to indicate the direction to move forward. DARPA Autonomous Vehicle (DAVE) [10] was the first attempt to use a Convolutional Neural Network (CNN) to extract road features to determine a vehicle's steering angle given an image from a forward-facing camera. NN-based BC was revitalized using DNN after deeper neural networks became viable options due to the advancement of parallel processing hardware, software (deep learning toolkits: Tensorflow and PyTorch), and algorithms such as batch normalization. A team from NVIDIA presented an end-to-end BC to learn how to steer a vehicle using a CNN [1].

In off-road agricultural environments, there has been active research on BC using End-to-End (E2E) learning for autonomous navigation. An autonomous row-following robot was proposed with a focus on low cost, and no extensive re-engineering [2]. They used a single front camera to infer steering angles for an autonomous row-following. A Deep CNN (DCNN) was pre-trained with open datasets and deployed their robot. Agronav is an autonomous navigation framework for agricultural robots [7]. This framework is an E2E vision-based but uses semantic segmentation and line

[1]Feeza Khan Khanzada and Jaerock Kwon are with Electrical and Computer Engineering, University of Michigan-Dearborn, 4901 Evergreen Road, Dearborn, Michigan 48128, USA {feezakk, jrkwon}@umich.edu

[2]Bada Kwon is with Computer Science, Northeastern University, 360 Huntington Ave, Boston, MA 02115, USA kwon.ba@northeastern.edu

[3]Woojin Jeong is with WApplE Cloud Co., Ltd., 313 Teheran-ro Gangnam-gu, Seoul 06151, Korea woojin@wapplecloud.com

[4]Young Seek Cho is with Electronic Engineering, Wonkwang University, 460 Iksan-daero, Iksan-si, Jeollabuk-do 54538, Korea ycho@wku.ac.kr

detection. Despite the successes of these various approaches, few analytical studies have been conducted on ROI and dataset size with respect to the performance of the trained DNN-based controllers.

III. Method

A. Operational Environment

The simulation environment employed in this study comprises two fundamental components: the robotic platform and the agricultural landscape. Specifically, we utilized AgileX Robotics' Scout 2.0 robot as the robotic platform [11]. For simulating the agricultural environment, we utilized an orchard farm environment [12]. This simulated world has many rows of trees separated by dirt paths, and the terrain is not flat. Small bumps in the dirt paths add an additional challenge to vision-based navigation because the shape of off-road terrain changes dramatically. We added concrete barriers to the original environment to prevent a robot from getting out of the environment. To add variations to the robot's viewpoint, oak trees, and a parked vehicle were also added. We used the Robot Operating System (ROS) [13] Noetic version in Ubuntu 20.04. The source code for the robot and its environment is open and managed at a GitHub repository [14].

B. Data Collection

The dataset for the DNN-based BC can be formulated as:

$$D = \left\{o(i), a(i)\right\}_{t=1}^{T}, \tag{1}$$

where $o(i)$ is the observation collected at the time i, and $a(i)$ is the corresponding actions (i.e., steering, throttle, brake) at that particular time, T represents the total time step of the data collection. Observations include front camera input images, the velocities, accelerations, and locations of a robot. Actions are steering, throttle, and brake of a robot. In this study, we used the input images from a front camera as an observation and steering angles as an action.

We collected data in three different conditions: the default simulated setting, different lighting directions, and different ground colors. The changes in tree shadow directions and lengths represent daily variations of the simulated environment. The ground color changes indicate a seasonal variation.

To show the effect of the training dataset size, we prepared two sets of data: Full and Half. *Full dataset* represents the entire data, including the default simulated, different lighting directions, and different ground colors. *Half dataset* comprises of randomly selected 50% of *Full data*. This amalgamation resulted in a dataset comprising a total of 119,052 images, each accompanied by corresponding labels for steering angle, throttle, and brake, as well as robot orientation and position. Subsequently, the full dataset was randomly partitioned into two equal subsets, each containing 59,526 images to constitute *Half dataset*.

In the collected dataset, there will be dominant zero steering angles since the robot needs to move forward in most cases. Thus, balancing the dataset with respect to steering angles is required to train the neural network. We call this pre-processing *data normalization*. Around 70% of front camera images are mapped to near zero steering angle in this study. We normalized the dataset not to have more than 1,000 samples per each around 7.2° steering angle range. Note that the angle range is -90° to 90° (from the far right to the far left). This normalization ensures that a DNN does not learn to generate the zero steering angle to minimize the average error of steering angle prediction regardless of the changes in an input image.

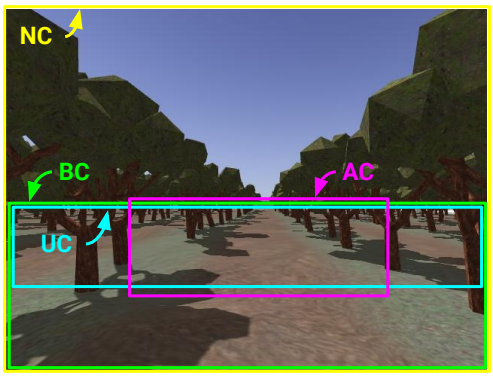


Fig. 1: Region of Interests in the input image. The entire image is NC (No Crop) - colored yellow, BC (Bottom Half Crop) is the bottom half of the image - green, UC (Upper part of the BC) is the upper half of the BC - cyan, and AC (Area Crop) is the center area - magenta.

C. Region of Interest

We selected four different ROIs to investigate the impact of different ROIs in the input image on the efficiency of DNN-based BC for E2E driving. See Fig. 1 for more details.

- NC (No Crop): The input image remained at its original resolution of 640×480 pixels, without any cropping.
- BC (Bottom Half Crop): The input image was cropped to 640×240 pixels, comprising only the bottom half of the original image.
- UC (Upper part of the BC) The input image was cropped to 640×120 pixels, containing the upper-bottom half of the original image.
- AC (Area Crop): The input image was cropped to a size of 400×130 pixels, representing a specific selected pixel area deemed optimal for network training.

By training the DNN-based BC model on each of these ROI configurations with both the full and half datasets, we aimed to evaluate how different regions of the input image affect the DNN model's performance in E2E driving tasks. This systematic investigation will provide insights into the optimal ROI for network training, ultimately enhancing the DNN model's efficiency and effectiveness in real-world driving scenarios.

D. Neural Network

We designed a DNN to extract important features from input images to infer corresponding steering angles. The DNN architecture for this study is inspired by the NVIDIA

PilotNet [1]. The architecture consists of four convolutional layers, each followed by a corresponding max-pooling layer, in addition to four dense layers. The first layer, acting as a lambda layer, is responsible for scaling the input image pixel data from the range of (0, 250) to (-1, 1). Conv2D layers are utilized to extract important features relevant to the steering angle inference task. The kernel size in all convolutional layers is three. The Rectified Linear Unit (ReLU) activation function was chosen and applied across all network layers except for the output layer. MaxPooling2D layers was used to downsample the convolutional layers. Finally, a flattened layer precedes three Dense layers, gradually decreasing the node size and culminating in a single output representing the steering angle. The Mean Squared Error (MSE) loss function is employed to evaluate the disparity between the predicted steering angle and the ground truth.

DNN-based BC learns from expert demonstrations. Thus, the training can be formulated as finding the optimal policy π with parameters θ using the following equation:

$$O \rightarrow \text{policy} : \pi_\theta \rightarrow A, \tag{2}$$

where $O = \left\{o(i)\right\}_{t=1}^{T}$, and $A = \left\{a(i)\right\}_{t=1}^{T}$. The pairs of ($O$, A) will be provided a the form of (1). The policy π is implemented with a DNN and θ is a set of connection weights of the DNN.

The work for the data collection, configurations of ROIs, and DNN-related efforts are open-sourced and managed at a GitHub repository [15].

E. Performance Metrics

The performance metrics that we used to assess the effectiveness of the selected techniques include driving deviation, completion time, and human intervention, which are inspired by the Online Performance Evaluation Metrics Index (OPEMI) [8].

1) Driving Deviation: Driving deviation is quantified as the sum of all the distance values calculated by using (3) from the desired path to the robot's current location during navigation in one driving session. The number of data points where a distance was measured can be different in a lane. So, to balance the number of data points between trials, we selected 1,000 data points from each navigation trial in a lane.

$$\text{Driving Deviation} = \sum_{i}^{N} \frac{\mid Ax_i + By_i + C \mid}{\sqrt{A^2 + B^2}}, \tag{3}$$

where i is an index of the robot's location, N is the total number of the robot's location data points, (x_i, y_i) represents a location of the robot, and A, B, and C are coefficients defining a straight line ($Ax + By + C = 0$) representing the desired path.

2) Completion Time: Completion time denotes the duration taken by a robot to travel from the starting point to the target within lanes. This duration includes instances where humans intervene to assist the robot. The time does not stop during the rescue session by a human.

3) Autonomy: Autonomy characterizes the robot's capability to achieve its target with less human assistance. Human interventions occur when the robot encounters obstacles like trees or puddles. Autonomy can be calculated using (4), inspired by the autonomy metric proposed by [1]. In our adaptation, we penalize human interventions with a duration of 15 seconds instead of the original 6 seconds, as it typically takes about 15 seconds for a human to readjust the robot's pose in an orchard farm setting.

$$\text{Autonomy} = \left(1 - \frac{N \times 15\ (seconds)}{T(seconds)}\right) \times 100, \tag{4}$$

where N is the number of human interventions, and T is the elapsed time the total time the robot navigates from the starting point to the target.

IV. Experiments

We trained DNNs with two datasets (*Full* and *Half dataset*) with four different ROIs (NC, BC, UC, and AC. See Fig. 1), which constitutes eight neural network controllers. Their navigation performance was tested with the proposed metrics at five different lanes in the simulated orchard farm. These five lanes are depicted in Fig. 2 Our experiments involved navigating the robot through five distinct lanes, labeled as lanes 1 to 5, as illustrated in Fig. 2. We used all eight DNN controllers to have the robot autonomously drive each lane twice for comprehensive evaluation purposes. Initially, the robot was tasked with moving from its starting position (assumed to be the origin of the lane) to the end of the lane. Upon reaching the designated endpoint, the robot was instructed to navigate back to its initial position within the same lane. These navigation tasks were referred to as left-to-right (LR) and right-to-left (RL) movements, corresponding to the respective directions of travel within the lane. We had the robot autonomously navigate both LR and RL in the five lanes and collected all the waypoints of the robot driven by a DNN. The driving trajectories, along with additional information such as timestamps, the robot's poses, velocities, and accelerations, were used in the performance evaluations. Keras framework 2.14.0 with TensorFlow 2.15 was used to train, test, and evaluate the DNNs. The DNNs were operated within a Python 3.9 environment on Ubuntu 20.04 with the ROS Noetic distribution. Note that Python 3.10 was not compatible with ROS Noetic. An NVIDIA GeForce RTX 2080 GPU with 8 GB RAM was used to accelerate the DNN training and inference. The main CPU was the Intel Core i9-9900, and the system memory size was 32 GB.

V. Results

The min-max normalized performance evaluation results are shown in Table I. All four different ROIs with two different datasets were tested in Driving Deviation (DD), Completion Time (CT), and Autonomy (AT). We put a tilde to DD and CT to indicate the normalized values: $\widetilde{DD}$ and $\widetilde{CT}$. The raw values of these metrics and actual driving paths are reported in the Appendix. Table I provides a

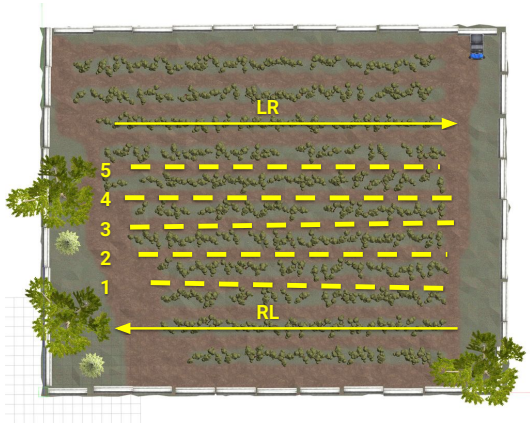


Fig. 2: A bird-eye view of the orchard farm environment in the simulator. Dashed lines with a number represent the lanes that the robot was tested for navigation. Yellow arrows indicate navigation directions: Left-to-Right (LR) and Right-to-Left (RL).

comprehensive summary of key evaluation metrics pertaining to navigation in both LR and RL directions. All numbers in the table are average values from driving five selected lanes (Fig. 2).

TABLE I: Normalized performance evaluations

Dataset	ROI	$\widetilde{DD}\downarrow$		$\widetilde{CT}\downarrow$		$AT\%\uparrow$	
		LR	***RL***	***LR***	***RL***	***LR***	***RL***
Full	NC	1.00	1.00	1.00	0.98	**100.00**	94.40
	BC	**0.00**	0.04	0.85	1.00	88.70	88.87
	UC	0.05	0.05	**0.00**	**0.00**	**100.00**	**100.00**
	AC	0.03	**0.00**	0.77	0.28	**100.00**	**100.00**
Half	NC	0.39	0.45	0.46	0.43	87.70	74.80
	BC	0.05	0.02	0.24	0.29	**100.00**	**100.00**
	UC	0.06	0.05	0.25	0.30	**100.00**	**100.00**
	AC	0.04	0.04	0.23	1.00	54.49	72.18

Firstly, we analyzed the driving deviation, which did not exhibit significant improvements or deterioration across the different techniques. However, the deviation is comparatively greater when the model is trained with full sized image in both half and full sized dataset. Additionally, DD and CT are incorporated to ensure the consistency and reliability of the obtained results. Our results reveal differences in neural network performance based on the selection of an ROI and dataset size. Full-sized images induce greater driving deviation, while cropped Regions of Interest (ROIs) show comparable or superior performance, highlighting the efficacy of selective visual focus in driving simulation environments. These results emphasize the importance of optimizing image size for improved driver training and road safety measures. Furthermore, prioritizing a cropped ROI over full-sized input images demonstrates improved *Autonomy*, enabling the network to allocate computational resources more efficiently to relevant environmental cues.

We have contrasted the efficacy of dataset size and ROI through performance metrics outlined in Section III-E. Various scenarios may prioritize one metric over the other, as articulated in the following equation formulation.

$$P_\omega = \omega_{DD}(1-\widetilde{DD})+\omega_{CT}(1-\widetilde{CT})+\omega_{AT}(AT/100), \quad (5)$$

where ω_{DD}, ω_{CT}, and ω_{AT} are weight variables to prioritize the precision driving (by using normalized driving deviation ($\widetilde{DD}$)), faster completion time (by using normalized completion time ($\widetilde{DD}$)), and autonomy (by using the autonomy metric AT) over one another in different scenarios. Note that $\omega_{DD}+\omega_{CT}+\omega_{AT}=1$. The weights can be determined based on use cases. Three sample scenarios are shown in Table II: Precision, Speed, and Autonomy. These scenario-based results show that (i) using a bigger dataset is better regardless of ROI choices, and (ii) UC is a good ROI regardless of the dataset size in the three different scenarios.

TABLE II: Overview of the scenarios where weighted performances based on three different cases

Scenarios	$\omega_{DD},\omega_{CT},\omega_{AT}$	Dataset	ROI	$P_\omega\uparrow$
Precision	0.85, 0.075, 0.075	Full	NC	0.07
			BC	0.93
			UC	**0.96**
			AC	0.92
		Half	NC	0.62
			BC	0.94
			UC	0.93
			AC	0.86
Speed	0.075, 0.85, 0.075	Full	NC	0.07
			BC	0.27
			UC	**1.00**
			AC	0.34
		Half	NC	0.57
			BC	0.79
			UC	0.78
			AC	0.11
Autonomy	0.075, 0.075, 0.85	Full	NC	0.85
			BC	0.84
			UC	**1.00**
			AC	0.94
		Half	NC	0.83
			BC	0.98
			UC	0.98
			AC	*0.54*

VI. Conclusion

This study highlights the significant impact of an ROI selection and dataset size on DNN-based BC controller performance for autonomous navigation. By considering these factors, designers can tailor network architectures to effectively address precision, speed, and autonomy requirements in navigation tasks. Future studies may explore advanced techniques for dynamically adjusting an ROI and dataset size based on real-time navigation conditions, further enhancing the adaptability and robustness of autonomous navigation systems. We plan to explore further by applying different network architectures and more fine-grained dataset sizes and deploying them to an actual robot in a real physical agricultural environment.

Acknowledgment

This study is supported by the Ministry of Education and National Research Foundation of Korea through the "Leaders in INdustry-university Cooperation (LINC) 3.0" Project.

REFERENCES

[1] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to End Learning for Self-Driving Cars," Apr. 2016. [Online]. Available: http://arxiv.org/abs/1604.07316

[2] M. Bakken, R. J. D. Moore, and P. From, "End-to-end Learning for Autonomous Crop Row-following∗," *IFAC-PapersOnLine*, vol. 52, no. 30, pp. 102–107, Jan. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896319324218

[3] P. Biber, U. Weiss, M. Dorna, and A. Albert, "Navigation System of the Autonomous Agricultural Robot " BoniRob " *." [Online]. Available: https://www.semanticscholar.org/paper/Navigation-System-of-the-Autonomous-Agricultural-%E2%80%9C-Biber-Weiss/e85b58686704f355f1d755bb4757a3e1909e4e78

[4] W. Winterhalter, F. Fleckenstein, C. Dornhege, and W. Burgard, "Localization for precision navigation in agricultural fields—Beyond crop row following," *Journal of Field Robotics*, vol. 38, no. 3, pp. 429–451, 2021, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21995. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21995

[5] Y. Bai, B. Zhang, N. Xu, J. Zhou, J. Shi, and Z. Diao, "Vision-based navigation and guidance for agricultural autonomous vehicles and robots: A review," *Computers and Electronics in Agriculture*, vol. 205, p. 107584, Feb. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169922008924

[6] J. Shi, Y. Bai, Z. Diao, J. Zhou, X. Yao, and B. Zhang, "Row Detection BASED Navigation and Guidance for Agricultural Robots and Autonomous Vehicles in Row-Crop Fields: Methods and Applications," *Agronomy*, vol. 13, no. 7, p. 1780, Jul. 2023, number: 7 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2073-4395/13/7/1780

[7] S. K. Panda, Y. Lee, and M. K. Jawed, "Agronav: Autonomous Navigation Framework for Agricultural Robots and Vehicles using Semantic Segmentation and Semantic Line Detection." IEEE Computer Society, Jun. 2023, pp. 6272–6281. [Online]. Available: https://www.computer.org/csdl/proceedings-article/cvprw/2023/024900g272/1PBwX2TioSI

[8] D. Kim, A. Khalil, H. Nam, and J. Kwon, "OPEMI: Online Performance Evaluation Metrics Index for Deep Learning-Based Autonomous Vehicles," *IEEE Access*, vol. 11, pp. 16 951–16 963, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10047917

[9] D. Pomerleau, "ALVINN: An Autonomous Land Vehicle In a Neural Network," in *Proceedings of Advances in Neural Information Processing Systems 1*. Morgan Kaufmann, 1989, pp. 305–313.

[10] "DAVE: Autonomous Off-Road Vehicle Control Using End-to-End Learning | BibSonomy." [Online]. Available: https://www.bibsonomy.org/bibtex/20e9a839118a9283db3986eb4723693a3/idsia

[11] "AgileX Scout Unmanned Ground Vehicle 2.0." [Online]. Available: https://www.trossenrobotics.com/agilex-scout-unmanned-ground-vehicle-2-0.aspx

[12] "Clearpath Additional Simulation Worlds," Jan. 2024, original-date: 2020-06-26T17:11:05Z. [Online]. Available: https://github.com/clearpathrobotics/cpr_gazebo

[13] "ROS: Home." [Online]. Available: https://www.ros.org/

[14] J. Kwon, "AgriBot: Intelligent Agricultural Robot," Nov. 2023, original-date: 2023-10-31T21:16:18Z. [Online]. Available: https://github.com/jrkwon/agribot

[15] ——, "ROS Packages for Intelligent Agricultural Robot," Nov. 2023, original-date: 2023-05-01T01:33:58Z. [Online]. Available: https://github.com/jrkwon/agribot_ros

APPENDIX

TABLE III: Performance evaluation of each dataset for the respective region of interests

Dataset	**Region of Interest**	**Driving Deviation (DD) (meters)**		**Completion Time (CT) (seconds)**		**Autonomy**	
		LR	***RL***	***LR***	***RL***	***LR***	***RL***
Full	NC	1406.33 (±1296.84)	1426.38 (±1185.12)	273.59 (±20.91)	268.15 (±13.64)	**100.00% (±0)**	94.40% (±1)
	BC	**179.21 (±60.73)**	237.76 (±101.7)	265.33 (±17.64)	269.63 (±12.05)	88.70% (±2)	88.87% (±2)
	UC	237.11 (±86.24)	244.76 (±84.75)	**218.06 (±65.7)**	**214.31 (±62.21)**	**100.00% (±0)**	**100.00% (±0)**
	AC	220.87 (±105.92)	**186.34 (±82.34)**	260.76 (±18.34)	229.65 (±12.68)	**100.00% (±0)**	**100.00% (±0)**
Half	NC	655.13 (±108.89)	740.44 (±200.59)	243.9 (±132.54)	238.15 (±17.84)	87.70% (±2)	74.80% (±4)
	BC	235.21 (±94.8)	216.5 (±71.76)	231.26 (±15.2)	230.14 (±13.72)	**100.00% (±0)**	**100.00% (±0)**
	UC	258.37 (±64.67)	253.15 (±52.77)	231.78 (±15.1)	231.13 (±13.04)	**100.00% (±0)**	**100.00% (±0)**
	AC	231.08(±124.08)	237.76 (±101.7)	230.69 (±82.79)	269.63 (±12.05)	54.49% (±7)	72.18% (±5)

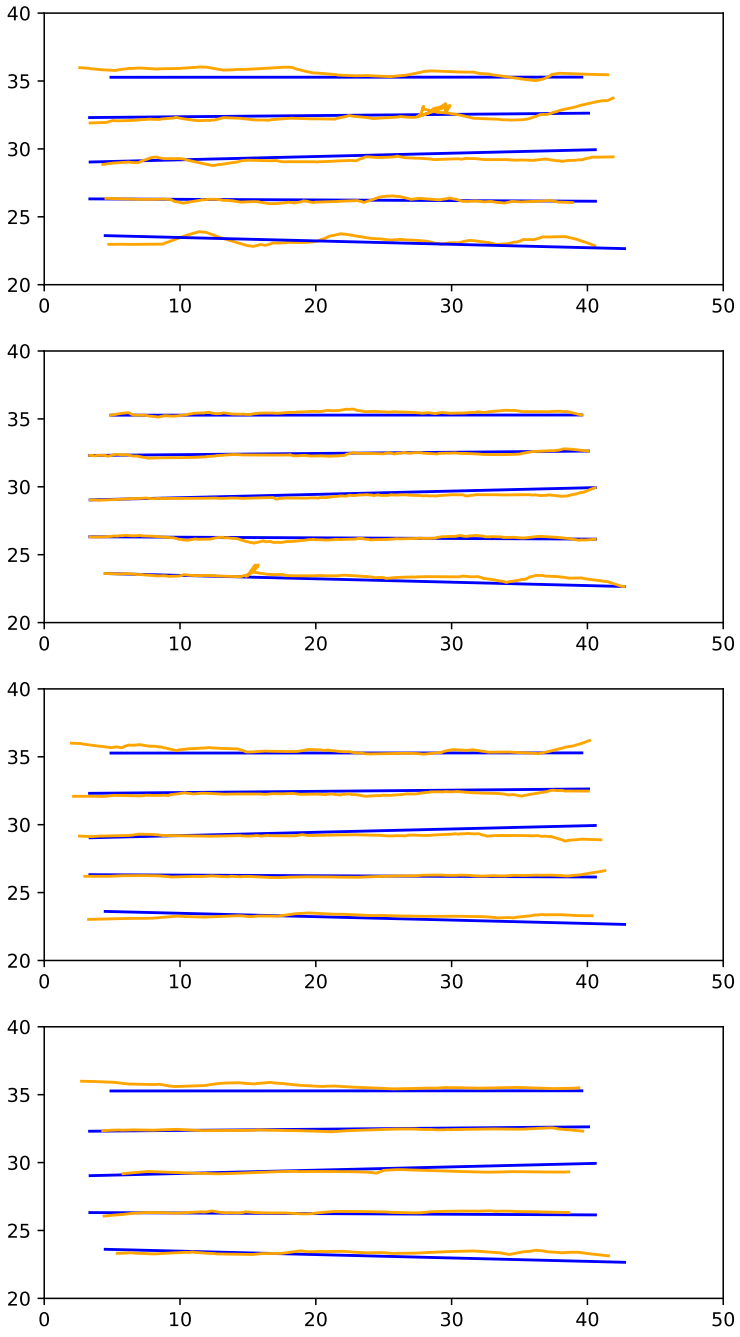


Fig. 3: LR (Left to Right) driving deviation from a straight line in each selected lane with *Full dataset* for NC, BC, UC, and AC from the top.

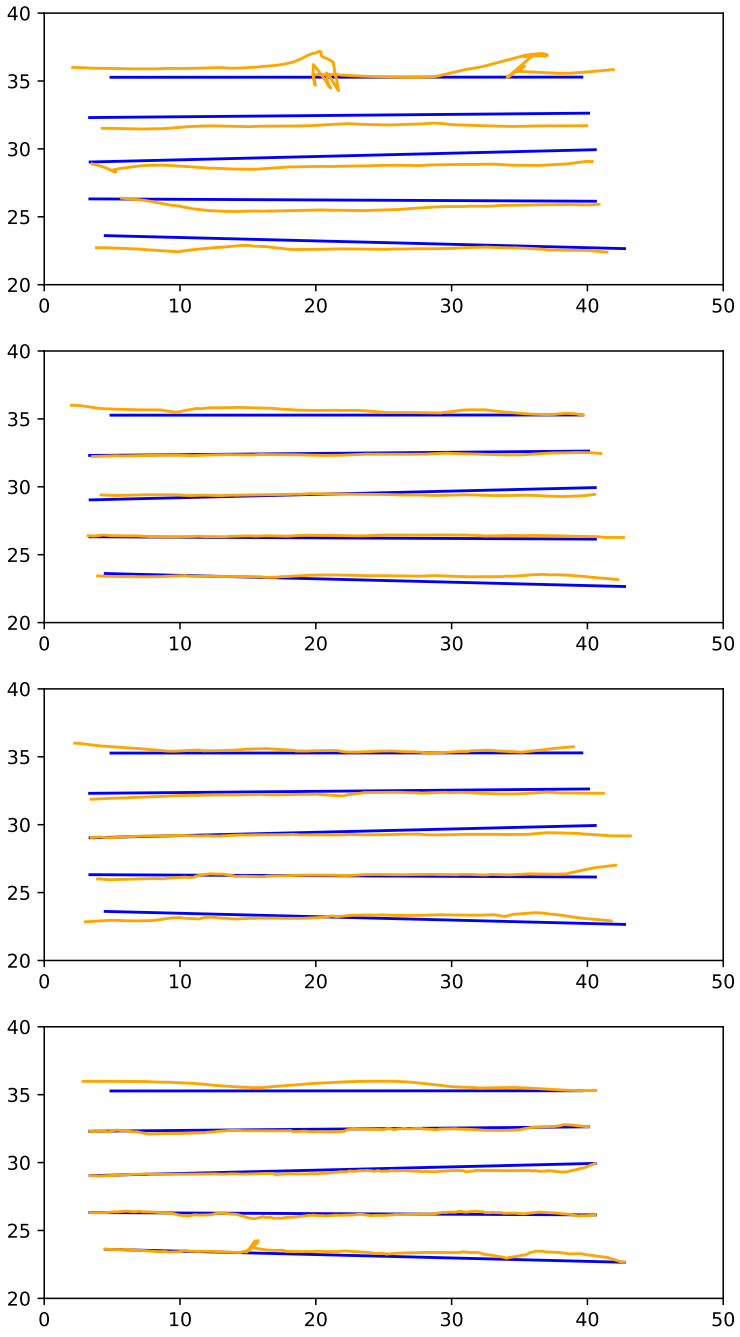


Fig. 4: LR (Left to Right) driving deviation from a straight line in each selected lane with *Half dataset* for NC, BC, UC, and AC from the top.

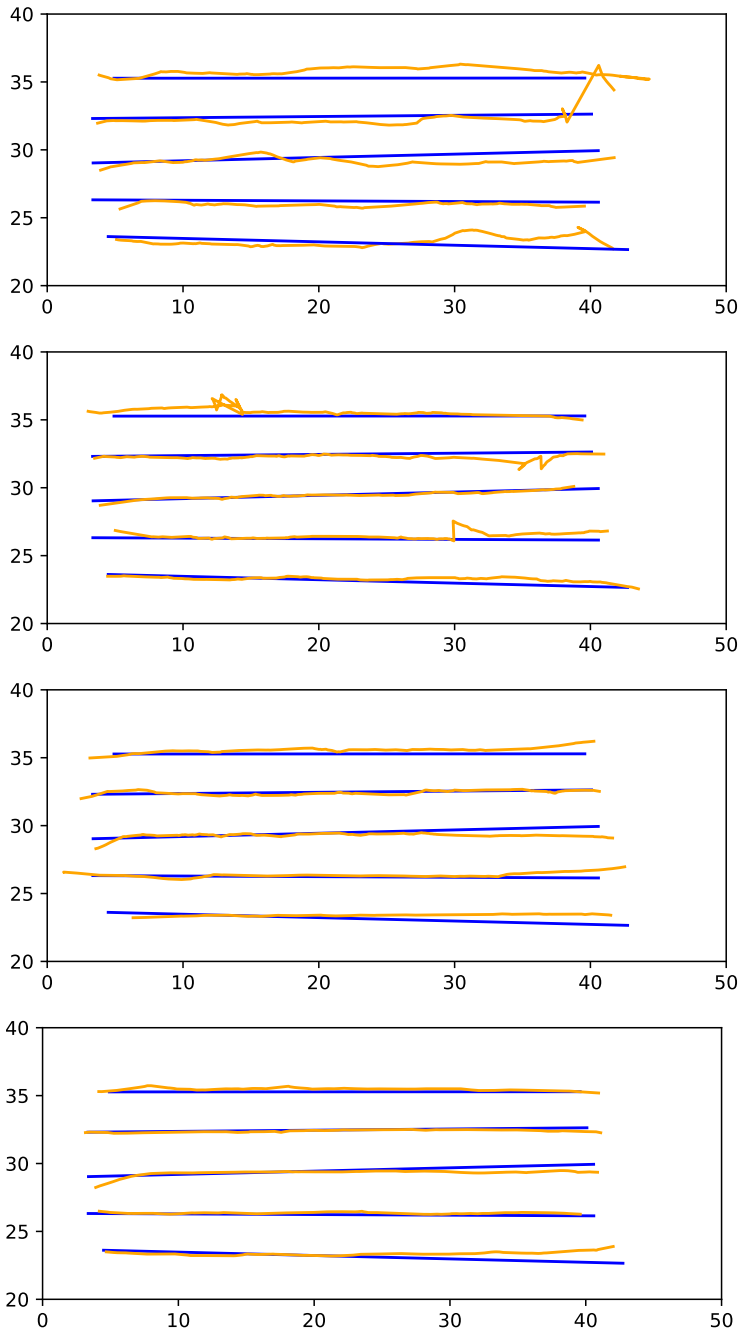


Fig. 5: RL (Right to Left) driving deviation from a straight line in each selected lane with *Full dataset* for NC, BC, UC, and AC from the top.

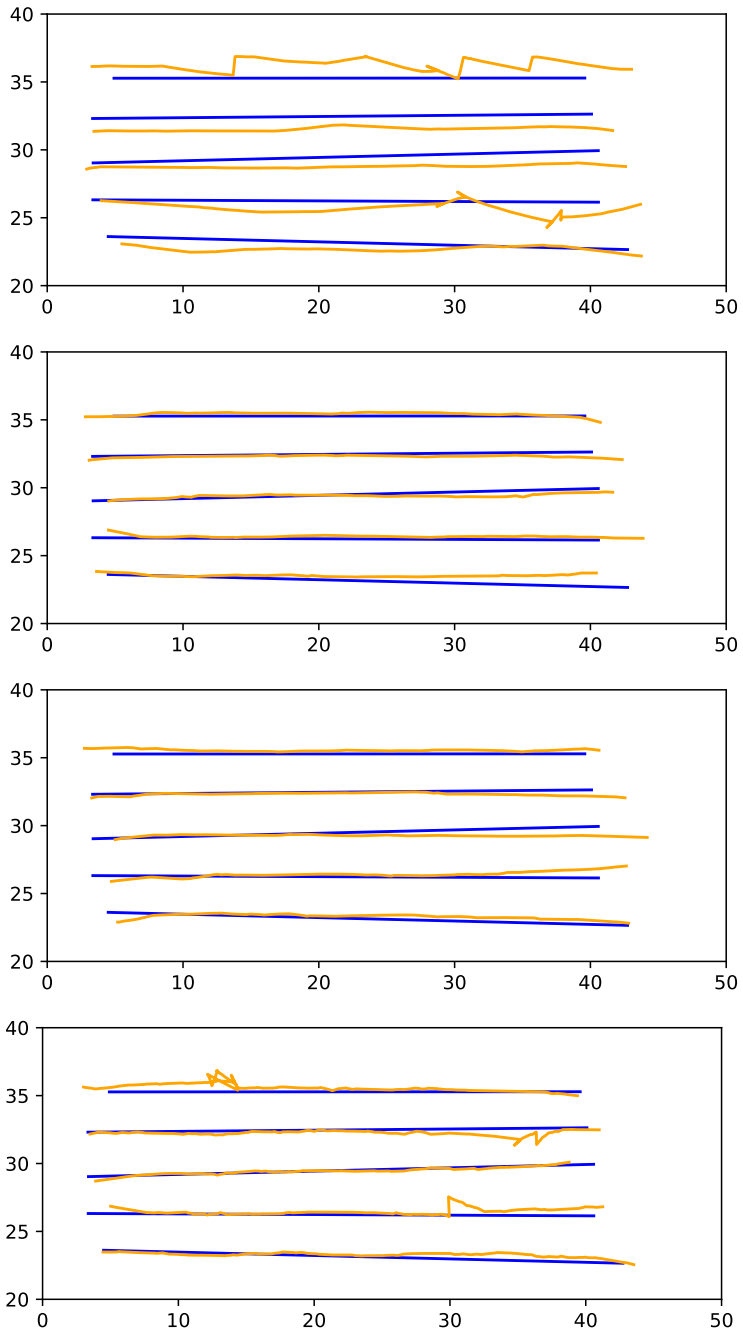
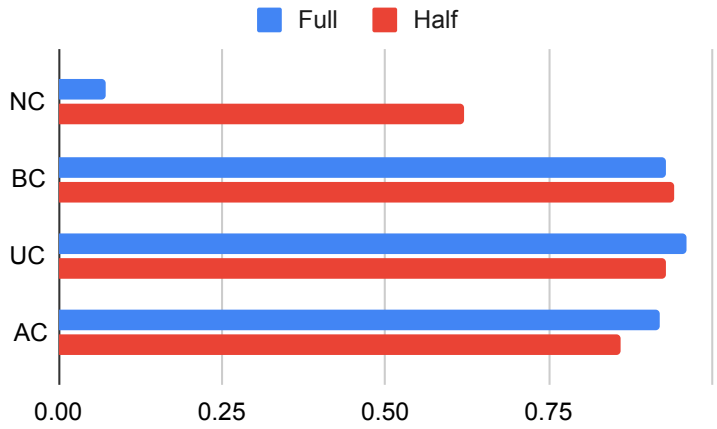
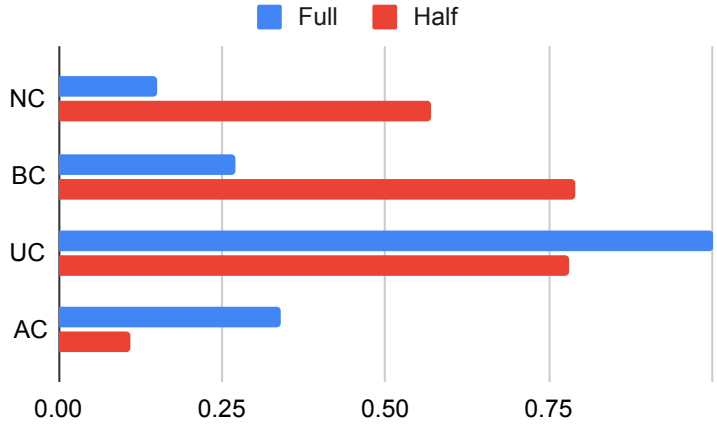


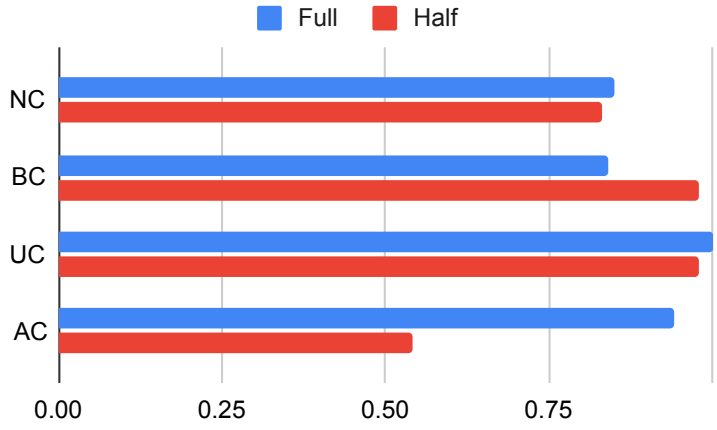
Fig. 6: RL (Right to Left) driving deviation from a straight line in each selected lane with *Half dataset* for NC, BC, UC, and AC from the top.



(a) Precision

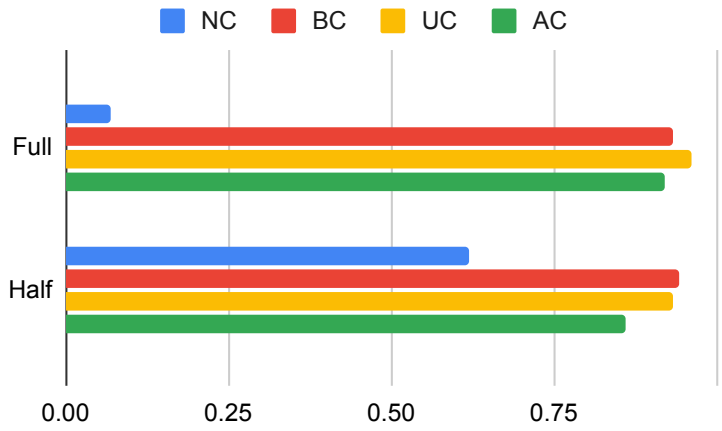


(b) Speed

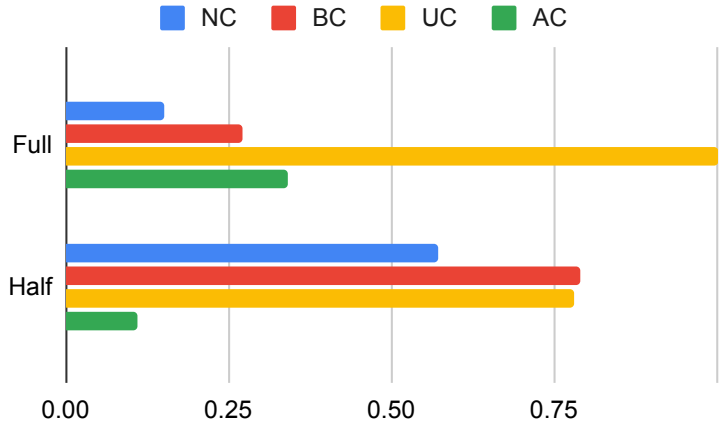


(c) Autonomy

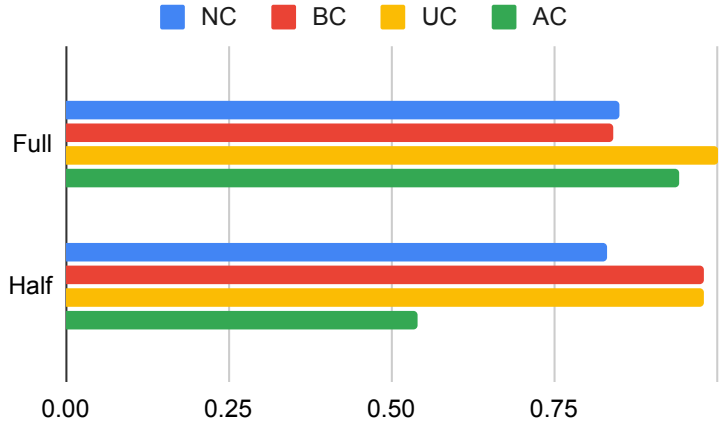
Fig. 7: Weighted performance of different ROIs with respect to the data size.



(a) Precision



(b) Speed



(c) Autonomy

Fig. 8: Weighted performance of different data sizes with respect to the ROIs.