IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 31 December 2023, accepted 2 February 2024, date of publication 6 February 2024, date of current version 14 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3362646

RESEARCH ARTICLE

RPRP-SAP: A Robust and Precise ResNet Predictor for Steering Angle Prediction of Autonomous Vehicles

SUMMRA SALEEM[1,2,3], MUHAMMAD NABEEL ASIM[1], LUDGER VAN ELST[1], PETER SCHICHTEL[2], AND ANDREAS DENGEL[1,3]
[1]German Research Center for Artificial Intelligence (DFKI), 67663 Kaiserslautern, Germany
[2]IAVGmbH, 10587 Berlin, Germany
[3]Department of Computer Science, RPTU Kaiserslautern-Landau, 67663 Kaiserslautern, Germany

Corresponding author: Summra Saleem (summra.saleem@dfki.de)

ABSTRACT Steering angle controller is a core module of autonomous vehicles, where a slight miscalculation can cause severe accidents. Following safety precautions, development of robust and precise steering angle predictor is an active area of research. However, existing simulator based steering angle predictors lack in predictive performance and have not been evaluated on same benchmark datasets. Furthermore, most of them are evaluated on simulated datasets and their potential on real-world data as well as in cross-domain evaluation of both types data (real-world, simulated) remain unexplored. To accelerate and expedite research related to steering angle prediction contributions of this paper are manifold: 1) It presents two benchmark datasets that are developed using Udacity and CARLA simulators. 2) Following the need for comparative study, over both simulated datasets, it benchmarks the performance of existing predictors under 2 different evaluation settings namely: same-track and cross-track evaluation. 3) In cross-domain evaluation, it explores generalization potential of predictors by training predictors on simulated data and evaluating them on 2 real-world datasets and vice versa. 4) It presents a robust and precise steering angle predictor that utilizes skip connections for proper gradient flow among different convolutional layers. In same-track evaluation where predictors are trained and evaluated on same-track data, proposed predictor outperforms existing predictors by achieving least Mean Absolute Error (MAE) of 0.13, 0.19 and 0.065 over lake track, jungle track and CARLA based datasets, respectively. Similarly, in cross-track evaluation where predictors are trained on one track and are evaluated on other track data, once again proposed predictor outperforms existing predictors by producing average least MAE errors of 0.33 and 0.06 over Udacity and CARLA datasets, respectively. Over two real-world datasets, Sully Chen and Comma.ai, the proposed predictor demonstrates superior performance compared to existing simulator-based predictors, achieving the lowest MAE of 2.41 and 0.50, respectively.

INDEX TERMS Autonomous vehicles, deep learning, prediction, residual block, steering angle, udacity simulator, CARLA simulator, virtual environment.

I. INTRODUCTION

Artificial Intelligence (AI) based automated systems are boosting the efficiency and completion speed of mundane and repetitive tasks which require extensive human labor work [67]. The prime motivation behind AI-supported automation is the development of smart tools that require minimal or no human intervention for completion of diverse types of tasks related to several domains i.e. transportation [6], banking [4], education, surveillance [65], energy management [51], oil and gas production [15]. Exponential growth and success of AI approaches bring breakthroughs

The associate editor coordinating the review of this manuscript and approving it for publication was Yanli Xu.

by facilitating development of robots [17] and autonomous vehicles (AVs) [73]. AVs have manifold benefits such as higher traffic efficiency, a lower collision rate due to active network communication between different vehicles on the road [76] and facility to self-commute for visually and physically impaired people [49].

The working paradigm of autonomous vehicles completely relies on complex intelligent systems where a minor malfunction might endanger several lives on road [20], [60]. Major brands of automotive industry such as Tesla,[1] Waymo,[2] BMW,[3] Mercedes-Benz[4] and Ford[5] are keenly interested in utilizing the power of artificial intelligence approaches for making complex intelligent systems competent in driving vehicles even better than humans [77]. Furthermore, to make sure reliable driving capabilities of intelligent systems, autonomous vehicles require robust and precise deep learning methods and their extensive testing. However, considering a wide range of unexpected situations in real-world scenarios, (such as response in emergencies, climate factors, traffic jams and complex routes) thorough testing of AVs is expensive and time-consuming [8], [28]. This drives the necessity of computer-aided programs competent in performing extensive testing of AVs for all possible unsafe scenarios that AVs can encounter during their movements on different tracks [41].

To empower intelligent systems and their computer-aided testing, development of simulators and deep learning predictors are active areas of research [21], [58]. Motivation behind the development of simulators (Udacity [54] and CARLA [18]) is to drive vehicles in virtual environment that reflects real-world graphical environment such as wide as well as narrow tracks with several types of hurdles like trees, humans, other vehicles and different weather effects [9]. While driving vehicles in virtual environments, the simulators record data related to different modules of intelligent systems such as track lane detection [56], [63], traffic signal detection [71], pedestrian behavior analysis [27], [37] and steering angle prediction [61]. The prime objective of researchers is to utilize simulated data for the development of deep learning predictors competent in accurately predicting different parameters of autonomous vehicles. The simulators also provide an option to integrate trained predictors into complex intelligent systems for evaluating their capabilities by driving vehicles on different tracks.

The focus of paper in hand revolves around development of a robust and precise deep learning predictor capable of controlling steering angle of autonomous vehicles. In autonomous vehicles steering angle module of complex intelligent systems controls directions of AVs. In the marathon of developing more accurate predictors, the goal of researchers has been to utilize diverse types of neural strategies for development of deeper architectures that could learn informative patterns and improve the efficiency and robustness of steering angle predictor. However, during learning phase in deeper networks gradient cannot flow properly which leads towards exploding and vanishing gradient problems and predictor only extracts limited informative features. Most of existing simulator based steering angle predictors are primarily evaluated on the Udacity simulator. However, it's worth noting that the CARLA simulator provides a more comprehensive platform for developing simulated datasets with diverse types of graphics that closely resemble real-world scenarios. Udacity simulator provides a platform for developing two distinct track datasets: jungle track and lake track. Interestingly, existing predictors are typically evaluated in a cross-track setting, where they are trained on one track and then evaluated on the other track. Surprisingly, none of the existing predictor is evaluated on same-track dataset.

Furthermore, it's worth noting that only three predictors [1], [34], [52] are evaluated in a cross-domain setting involving both simulated and real-world datasets. Hence, the potential of predictors remains unexplored in cross-domain evaluation setting where predictors are trained on real-world data and are evaluated using simulated data and vice versa. Considering room for evaluating existing steering angle predictors [1], [3], [5], [13], [24], [50], [66], [70] from different perspectives, manifold contributions of this paper are summarized below:

- To accelerate and expedite research by developing more robust and precise predictors for steering angle controlling, it facilitates 2 public benchmark datasets that are developed using Udacity and CARLA simulators. With an aim to facilitate in-depth evaluation of predictors across same and cross-track settings, we created three distinct versions of the Udacity dataset: one based on the lake track, one based on the jungle track, and a combined version that includes data from both the jungle and lake tracks. Likewise, for the CARLA dataset, we developed six distinct versions: five versions corresponding to each of the five different towns and one comprehensive version that encompasses data from all five towns.
- Considering the need of existing predictor's performance comparison on same benchmark datasets, following working paradigm of existing predictors in their research articles, we implement 8 most recent simulator-based steering angle predictors and benchmark their performance in cross-track and same-track settings on newly developed datasets.
- To explore generalization and real potential of steering angle predictors, we evaluate predictors in cross-domain setting which allows to train predictors on 2 real-world datasets and evaluate them on two simulated datasets and vice versa.
- Following the success of dense architectures which provide alternative paths for proper gradient flow that facilitate extracting comprehensive features by avoiding

[1]https://www.tesla.com
[2]https://www.waymo.com
[3]https://www.bmw.de
[4]https://group.mercedes-benz.com/
[5]https://www.ford.com

vanishing and exploding gradient problems, We propose a unique predictor based on ResNet modules. Performance comparison of proposed predictor with existing predictors reveals proposed predictor produces better performance on both simulated and real-world datasets.

II. RELATED WORK

This section provides a brief overview of three distinct categories of predictors: those assessed solely on simulated data, those assessed solely on real-world data, and those evaluated on both simulated and real data.

A. SIMULATED DATA BASED PREDICTORS

According to the best of our knowledge in last 4 years, 8 different deep learning predictors [1], [3], [5], [13], [24], [50], [66], [70] have been proposed for empowering steering angle controllers by utilizing virtual environment of simulators. However, among 8 existing simulator based predictors [1], [3], [5], [13], [24], [50], [66], [70] work of 4 predictors [3], [5], [24], [66] is limited as authors did not perform comprehensive experimentation and evaluation. Despite these limitations, these predictors are included in this study to provide a detailed overview of existing steering angle predictors.

Among 8 existing predictors 6 studies [1], [5], [13], [24], [30], [50] have developed convolutional neural network (CNN) based predictors that predict steering angle by capturing features related to track and hurdles. Mohammadi et al. [50] developed steering angle predictor by utilizing traditional descriptors (Harris [29], SIFT [46], SURF [7], FAST [59] and BRIEF [11] to find important regions of images that contain more comprehensive information. Authors fed these extracted regions of interest to CNN architecture [10] for the extraction of more useful features. Furthermore, 4 predictors (Kalim et al. [13], Garg et al. [24], Hassan et al. [30] and Bayarov Ahmedov et al. [1]) use 5 convolutional layers for the extraction of deeper features while Anchalia and Srividhya [1], [5], [13], [30] predictor makes use of 3 convolutional layers. Ali [3] developed steering angle predictor by utilizing 4 CNN layers. In 4 different predictors [1], [13], [24], [30] although number of convolutional layers are similar, they slightly differ in other neural strategies such as Garg et al. [24] predictor uses dropout layer for avoiding model over-fitting and Bayarov et al. [1] predictor performs data normalization that facilitates convolutional layers for the extraction of more comprehensive features. Kalim et al. [13] predictor extract non-linear patterns of features by making use of additional rectified linear unit (relu) after each convolution layer. Hassan et al. [30] predictor utilizes max pooling layers to retain the most relevant features.

To reap the benefits of both convolutional neural network and recurrent neural network architectures Valiente et al. [70] proposed hybrid predictors. Convolution architecture extracts informative patterns and LSTM architecture [79] aids to incorporate temporal features of environment. Following the success of transformers [26] in diverse types of Natural Language Processing (NLP) tasks such as text classification [68], text summarization [42], machine translation [69] and question answering systems [55] Shvejan et al. [66] utilized 8 stacks of transformer architecture to develop steering angle predictor. Khan et al. [40] proposed steering angle predictor comprising two different modules. The first module generates low-dimensional latent semantic representation of the image. The second module is trained using reinforcement learning and utilized the latent vector as input to predict steering angle.

B. REAL-WORLD DATA BASED PREDICTORS

Mygapula et al. [53] utilized a normalization layer and 9 convolutional layers to design a a deep architecture for steering angle prediction. Motivation behind development of deeper architecture was to extract high-level semantic features. Saleem et al. [62] also designed steering angle predictive pipeline by utilizing convolution neural network and two metaheuristic algorithms i.e. bat and particle swarm optimizer. Metaheuristic algorithms takes a range of CNN layers, filters and hyper parameters and automatically find optimal values of hyper-parameters and number of layers. Ijaz et al. [35] developed a hybrid predictor that made use of U-Net and LSTM based architectures. U-Net architecture extracts high-resolution as well as Hierarchical features and spatial context features. LSTMs extracts Temporal dependencies and Sequential patterns. Du et al. [19] utilized pre-trained ResNet architecture along with 3DCNN and LSTM architectures to design powerful steering angle predictor. Motivation behind utilization of pre-trained ResNet architecture was to apply transfer learning such as trained ResNet architecture on large dataset and utilize model trained weights to further fine-tune on steering angle data. Ye et al. [78] proposed YOLOv5 based architecture for steering angle predictor. Primarily, the authors trained YOLOv5 on self generated dataset for lane detection. Afterwards, the output of the YOLOv5 is fed as input to subsequent CNN based steering angle predictor. A major drawback of this architecture is its reliance on YOLOv5 for feature extraction which can hinder performance of steering angle predictor. Oniar et al. [57] predictor made use of CNN architecture and multi-head attention for steering angle prediction. Wu et al. [74] proposed CNN and LSTM based hybrid network for steering angle prediction. CNN layers extract spatial features and LSTM layers aid to capture temporal dependencies which allows the model to effectively learn both sequential patterns and spatial relationships. Table 1 summarizes all three types of predictors in terms of year, approach and used datasets.

C. REAL-WORLD AND SIMULATED DATA BASED PREDICTORS

Munir et al. [52] predictor made use of CNN architecture and self attention layers. CNN layers extract discriminative features and attention layers focused on more informative features. Hou et al. [34] designed heterogeneous auxiliary

TABLE 1. A comprehensive summary of existing steering angle predictors.

Author	Approach	Training Dataset
Simulated Data based Predictors		
Mohammadi et al. 2023, [50]	Harris, SIFT, SURF, FAST, BRIEF, CNN	Udacity simulator
Hassan et al. 2022, [30]	CNN	CARLA simulator
Anchalia et al. 2022, [5]	CNN	Udacity simulator
Garg et al. 2022, [24]	CNN	Udacity simulator
Shvejan et al. 2021, [66]	Transformer encoder	Udacity simulator
Ali et al. 2021, [3]	CNN	Udacity simulator
Bayarov et al. 2021, [1]	CNN + RNN	Udacity simulator
Kalim et al. 2021, [13]	CNN	Udacity simulator
Real-world Data based Predictors		
Ye et al. 2022, [78]	YOLOv5	Self-generated dataset
Oinar et al. 2022, [57]	RGB backbone + optical flow + transformer encoder	Udacity
Saleem et al. 2022, [62]	Particle swarm optimizer and bat algorithm to optimize CNN	Udacity
Mygapula et al. 2021, [53]	CNN	Sully Chen
Ijaz et al. 2019, [35]	Unet+ DNNs+ LSTM	TAMUC, Udacity
Du et al. 2019, [19]	3D CNN + LSTM, ResNet	Udacity
Wu et l. 2019, [74]	CNN + LSTM	Udacity
Real-world + Simulated Data based Predictors		
Munir et al. 2022, [52]	Frame-based RGB encoder, event encoder and decoder	Davis Driving dataset (DDD), CARLA EventScape Dataset, Self-generated dataset
Hou et al. 2019, [34]	3D ResNet + LSTM + auxiliary networks	Udacity simulator, Comma.ai

networks feature mimicking that not only considers the steering direction but also incorporates comprehensive contextual information. However, each auxiliary network performs different yet related tasks such as image segmentation or optical flow estimation. Both predictors [34], [52] are evaluated across both real world and simulated data.

III. MATERIAL AND METHODS

This section summarizes working paradigm of proposed RPRP-SAP predictor and development process of benchmark datasets. It also describes evaluation measures and criteria used to verify the authenticity of proposed and existing predictors.

A. PROPOSED PREDICTOR

The proposed RPRP-SAP predictor comprises deep architecture having 1 convolutional layer and 4 ResNet blocks [31]. The prime motivation behind the design of deeper architecture is to extract more comprehensive and informative patterns from different tracks images. Although deeper architectures are well-known for extracting informative features, however, they are more prone to vanishing and exploding gradient problems. During back propagation, these problems adversely affect the optimization of learnable parameters of predictor. To avoid vanishing and exploding gradient problems, we enriched predictor architecture with shortcut connections known as skip connections [32] that

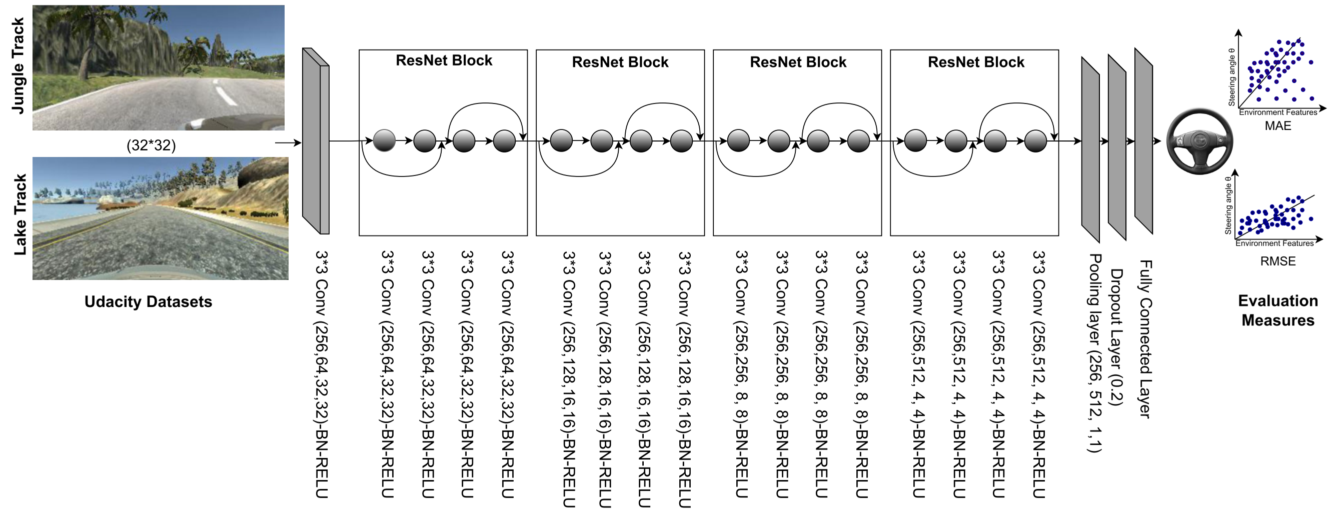


FIGURE 1. **Graphical illustration of proposed predictor. Here each ResNet block contains 4 CNN layers and in each block arrows represent to skip connection which provides alternative path for gradient flow. 3 × 3 Conv (256,64,32,32)-BN-RELU represents convolutional layer: 3 × 3 kernel, input channel (256) output Channels (64), image dimensions (32, 32), with batch normalization and RELU activation.**

facilitate optimal gradient distribution across different layers of network.

Figure 1 graphically illustrates the architecture details and key parameters of proposed RPRP-SAP predictor. Input image having 32 × 32 dimensions is passed to convolutional layer for the extraction of low-level representations of objects. Extracted representations after normalization are passed through relu that enrich low-level representation with non-linear patterns [25], [36]. The non-linear representation of images is further fed to 4 successive ResNet blocks that extract diverse types of patterns for accurate prediction of steering angle. Each ResNet block contains 4 convolutional layers in a sequential manner. Furthermore, features extracted through each convolutional layer are normalized [36] and enriched with non-linear patterns using relu activation function [25]. For proper gradient flow in each module, we utilize skip connections which utilize identity functions to create alternative paths between different layers of ResNet block.

It can be seen from Figure 1, the ResNet block allows extracted features to flow from one layer to next immediate layer and also to other next layers by skipping intermediate layers in between using identity function. To briefly understand the concept of identity functions, let's consider a CNN architecture that comprises L layers. At each layer, a composite function $H_L(\dot{})$ performs three different operations including convolution, normalization and relu activation. Existing steering angle predictors pass L^{th} layer output of composite function to $(L+1)^{th}$ layer. Mathematically, output of $(L+1)^{th}$ layer can be computed as shown in Equation 1.

$$x_{(L+1)} = H_L(x_L) \tag{1}$$

In above Equation, x_L and $x_{(L+1)}$ represents output of L^{th} and $(L+1)^{th}$ layer, respectively. Contrarily, in our proposed predictor identity function based output of $(L+1)^{th}$ layer can be mathematically written as follows:

$$x_{(L+1)} = H_L(x_L) + (x_L) \tag{2}$$

Equation 2 demonstrates that identity function concatenates output of x_L layer with output of composite function applied to x_L.

Furthermore, consecutive ResNet blocks of proposed predictor are designed in a manner to generate down-sampled image representation in each successive ResNet block. The process of up-sampling [80] involves increasing the spatial resolution [16] to highlight smaller regions of an image while maintaining its 2-dimensional representation [39]. Down-sampling ensures that predictor can process images at different scales and resolutions, enhancing its ability to extract meaningful features from various levels of detail in the input data. The extracted features are fed to average pooling [45] and dropout layers [22] to retain highly informative features and drop irrelevant and redundant features. To introduce regularization, proposed predictor randomly eliminates a few neuron connections of extracted features among hidden layers of network. Hence, neurons that are dropped neither participate during forward pass nor their weights are updated during backward pass. Output of ResNet blocks is passed to 3 fully connected layers [44] which extract global relations of extracted features and predict steering angle direction.

B. BENCHMARK DATASETS

Considering the need for public benchmark datasets in virtual environment, we utilize CARLA and Udacity simulators for the generation of 2 datasets. Following subsections briefly describe the process of datasets development, statistics of datasets and different types of pre-processing strategies that are utilized to enrich generated datasets with some graphical

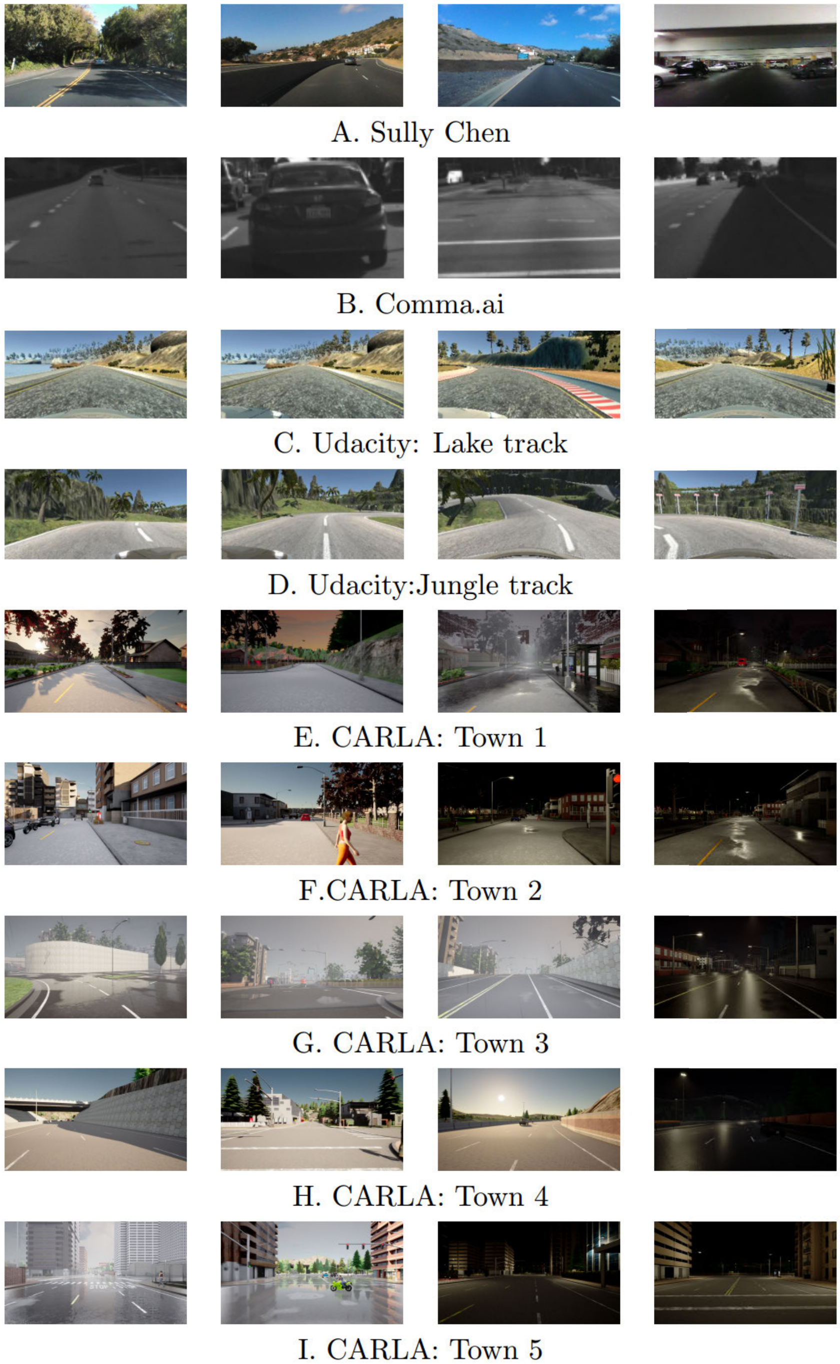


FIGURE 2. Sample images of simulated and real-world datasets.

characteristics that real-world datasets contain and may not exist in virtual environment of simulators.

1) UDACITY SIMULATOR

Udacity simulator [54] provides a virtual environment-based platform for developing and testing complex intelligent system modules of AVs in 2 distinct settings: training mode and autonomous mode. Training mode facilitates training data generation by manually running a vehicle on a particular track. During training mode, simulator captures images using 3 cameras mounted on dashboard of a vehicle. Moreover, it also provides information related to steering angle, throttle neck and speed of vehicle corresponding to all images. This training data is used to train and evaluate designed deep-learning predictors. In autonomous mode, simulator provides an option to integrate deep learning predictors into relevant modules. Furthermore, rather than manually controlling vehicles with keyboard, it utilizes deep learning predictors to drive vehicles. Specifically, it drives vehicles in virtual environment and users can analyze the working of a particular module that either developed deep learning predictors are properly working or not.

Udacity simulator provides 2 distinct tracks namely; lake and jungle tracks. The lake track is quite sunny and clear with only a few turns. In contrast to this, the jungle track has tight twisted turns and is surrounded by hills and trees which makes the track darker. The darker characteristic of jungle track throws shadow over the road resulting in complex driving track. Hence, considering the need for an efficient predictor for autonomous driving to address challenges of real-world environments, we generate 2 datasets from both environments with a total driving time of 1 hour for each track and a few sample images of the generated dataset for both tracks are shown in Figure 2.

2) CARLA SIMULATOR

CARLA simulator [18] is an open-source platform designed for development, validation and testing of autonomous driving systems. It facilitates a highly realistic virtual environment that enables researchers and developers to experiment with a variety of driving scenarios, vehicle behaviors and environmental conditions. It offers a wide range of features including: detailed urban environments, various vehicle models, traffic simulation and dynamic weather variations. Specifically, it provides 5 different urban towns with 14 different weather variations namely: clear noon, cloudy noon, wet noon, wet cloudy noon, mid rainy afternoon, wet sunset, soft rainy sunset, clear sunset, cloudy sunset, hard rain noon, soft rain sunset, clear night, cloudy night and default.

To benchmark performance of proposed and existing predictors, we generated 6 different versions of CARLA dataset including, town 1, town 2, town 3, town 4, town 5 and combined version. Table 2 demonstrates distribution of sample images into train, test and validation sets across all 6 versions of CARLA datasets. The generated data-set consists of driving video of approximately 50 minutes across each town. Furthermore, the generated dataset has dynamic weather configurations, which aids to test autonomous driving behaviour in various situations. The generated dataset comprises of RGB frames along with corresponding steering angle value. Figure 4 (B) graphically illustrates number of sample images corresponding to unique value of steering angle for CARLA dataset.

3) DATA PRE-PROCESSING

Considering the fact that data collected by driving vehicles on real-world tracks differs from data generated by driving vehicles in virtual environment. Although simulators are designed to generate synthetic data that reflects trends of real-world data while generating data the simulators lack in capturing significant factors that exist in real-world data recording. It can be seen from Figure 2 that real-world dataset differs significantly from simulated dataset. Furthermore, real-world data depicts a variety of objects including mountains, trees and other vehicles in diverse scenes, such as roads and parking lots with varying lightening conditions. To eliminate limitations of virtual environment-based data, we enriched generated data with essential factors of real-world data using diverse types of data pre-processing strategies that are briefly summarized below. The graphical illustration of pre-processing strategies across both tracks is shown in Figure 3.

- **Zooming:** Data generated through virtual environment lacks minute information about different objects of track. We included such information by zooming existing images in the range of 10% to 30%. This technique enables to focus on details of objects which are not captured with a normal camera angle.
- **Random flip:** The cameras mounted on dashboard of vehicle capture images at specific angles and offer little variety in scene. To introduce diverse types of angle rotations we randomly flipped images horizontally and vertically. This technique flips different objects of track upside down or sideways, to provide a diversity of scenes.
- **Brightness:** Data generated through simulator fails to reflect different lighting conditions on road tracks. Figure 2 illustrates unlike simulated data which is constrained to only one lighting scenario, real-world data has multiple lighting scenarios. Hence, to enrich data with images under varying illumination conditions, we randomly change the brightness of images with in the range of 10% to 30%. This generates data to depict varying light conditions in order to replicate changing weather conditions.
- **Color space transformation:** The colorization parameter of an image provides crucial information to intelligent algorithms for extracting informative features. Several image processing techniques aid to alter the colorization factor of images such as contrast [23], saturation [47], Hue [33] and YUV [72]. For this study, we have used YUV technique, which facilitates to adjust color space of images in order to highlight important regions with different color spaces.

4) STATISTICS OF DATASETS

The dataset generated through simulator contains diverse number of steering angles across both tracks. The number of unique steering angles in generated dataset of lake and jungle tracks is 39 and 41, respectively. Figure 4 (A) graphically illustrates number of sample images corresponding to unique value of steering angle for Udacity dataset. It is evident from Figure 4 (A) that data is highly imbalanced and more than 60 percent of sample images of generated data belong to a steering angle of "0" depicting straight steering direction. Hence, imbalanced data can lead to baisness in predictors towards a straight direction.

To overcome issue of imbalanced data, we first balance both dataset by dividing values of steering angle into 25 bins.

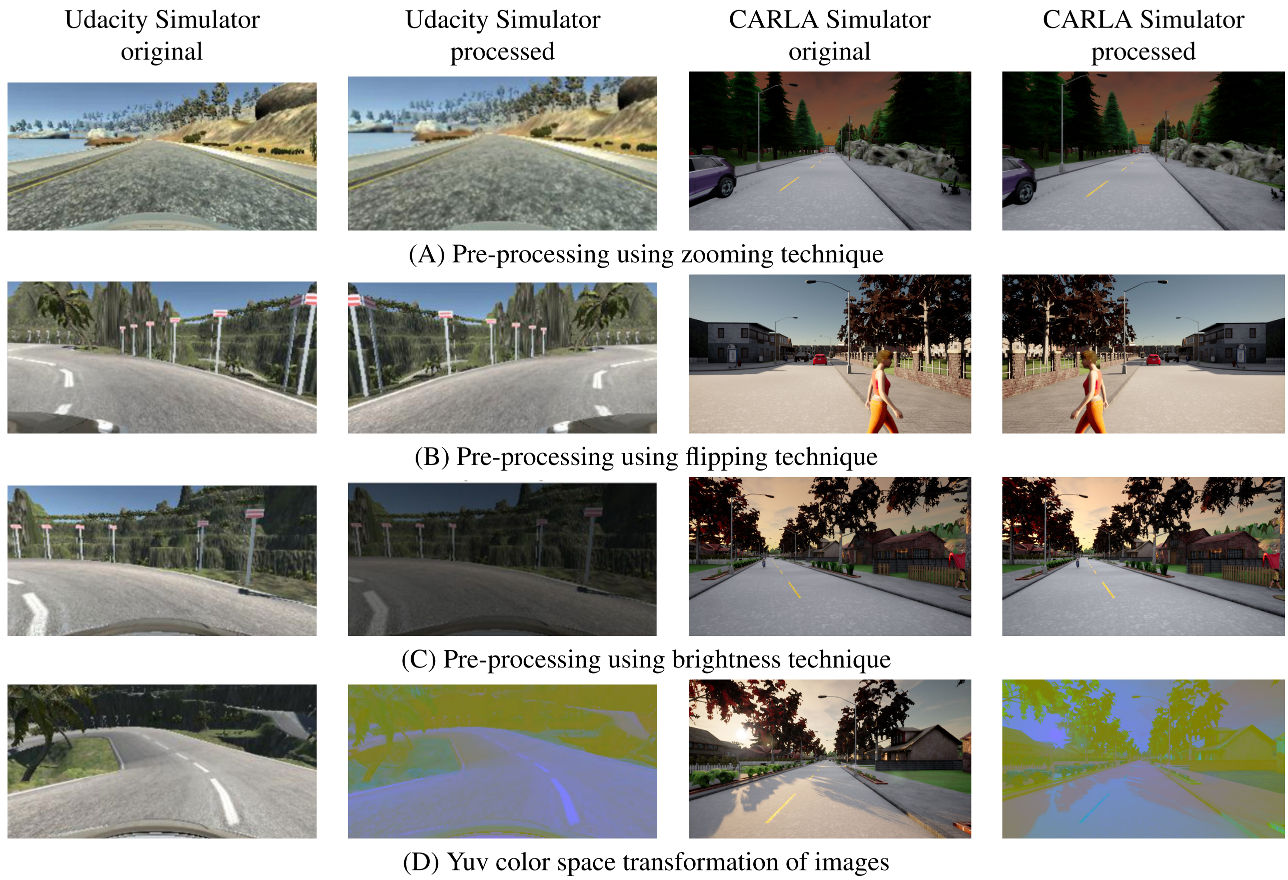


FIGURE 3. Data pre-processing techniques.

After converting values of steering angle into bins, we only retain 400 sample images in each bin and remove rest of sample images. Consequently, the pre-processed dataset is uniform in terms of number of sample images across all steering angles. Initially, a total number of 41,189 and 40,732 images are generated for lake and jungle tracks, respectively. However, after balancing the data number of sample images in lake and jungle tracks decreased to 14,691 and 26,727, respectively. Hence, a balanced version of data eliminates the tendency to predict biased value of steering angle. The sample images of both tracks are divided into train, validation and test sets with a split ratio of 70, 10 and 20, respectively. Table 2 provides an overview of the sample images training, validation and test sets for both the lake and jungle tracks individually, as well as for the combined Udacity dataset.

5) REAL-WORLD DATASETS

A large number of real-world datasets with diverse environments are available for steering angle prediction including Sully Chen [43], comma.ai [14] and Udacity. However, to compare and evaluate performance of proposed and existing predictors, we used a smaller dataset [43] comprising of around 45,000 images and a bigger dataset [14] consisting of approximately 400,000 images. In existing studies [1], [34], [53] comprehensive detail about both datasets is available, so here we only provided high level overview of both datasets.

TABLE 2. Statistics of simulated and real-world benchmark datasets.

Dataset	Train Set	Validation Set	Test Set
Comma.ai	371588	41288	103220
Sully Chen	31500	4500	9000
Udacity	29817	3315	8286
Jungle Track	19242	2139	5346
Lake Track	10575	1176	2940
Carla all Towns	174214	24198	43557
CARLA: Town 1	37737	5242	9435
CARLA: Town 2	32627	4532	8157
CARLA: Town 3	34984	4859	8747
CARLA: Town 4	32616	4531	8154
CARLA: Town 5	33882	4706	8471

1) Sully Chen: Sully Chen dataset was recorded during day time in 2017 using a Honda civic 2014 car around the area of san pedro and rancho palos verda California. The dataset comprises of approximately 45,500 images of frontal roads along with steering angles with a total size of 2.2 GB. Table 2 shows sample images of Sully Chen dataset. The dataset is divided into train,

Steering Angle (θ)	Number of Samples: Negative (θ)	Number of Samples: Positive (θ)
1	3696	2712
0.95	57	40
0.9	43	280
0.85	60	276
0.8	181	283
0.75	158	311
0.7	325	335
0.65	455	369
0.6	688	284
0.55	507	392
0.5	557	401
0.45	1002	477
0.4	961	371
0.35	837	499
0.3	245	383
0.25	952	473
0.2	2098	478
0.15	650	490
0.1	496	514
0.05	4	9
0	56972	

A. Udacity

Steering Angle (θ)	Number of Samples: Negative (θ)	Number of Samples: Positive (θ)
0.7	3464	5485
0.6	716	937
0.5	1415	1598
0.4	1786	19448
0.3	2322	2447
0.2	5632	5492
0.1	6997	7039
0	191401	

B. CARLA

FIGURE 4. Distribution of images across unique values of steering angles across Udacity and CARLA datasets.

validation and test set using a split ratio of 70%, 10% and 20%, respectively.

2) Comma.ai: The comma.ai is a publicly available dataset collected on a highway for 7.25 hours of driving, which is divided into 11 videos captured during both day and night time [14]. The dataset has several sensors that were measured in different frequencies. Following existing steering angle predictors [1], [34], this study also used camera frames with corresponding value of steering angle. Furthermore, to ensure a balanced ratio between day and night images, we split data into train, validation and test sets with a ratio of 70% (371588 samples), 10% (41288 samples) and 20% (103220 samples), respectively.

C. EVALUATION MEASURES

To evaluate the integrity and generalizability of proposed predictor, following evaluation criteria of existing studies, we utilize 2 different evaluation measures namely Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) [12], [38]. MAE calculates the difference between actual and predicted values followed by an absolute function in order to evaluate the predictor's performance, whereas mean square error (MSE) approximates the average difference between the square of ground truth and projected values. RMSE is calculated by taking square root of MSE value. The efficiency of the predictor will be improved by lower values for MAE and RMSE. Furthermore, if data fluctuates continuously the MAE and RMSE provide different values at different points. Equations 3 and 4 express mathematical formulations of MAE and RMSE, respectively.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|y_g^i - y_p^i\right| \tag{3}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_g^i - y_p^i\right)^2} \tag{4}$$

In Equations 3 and 4, y_g^i, represents actual value and y_p^i denotes predicted value for a sample i. N stands for the overall sample count.

IV. EXPERIMENTAL SETUP

The proposed predictor is developed on top of 8 different APIs namely; pytorch,[6] numpy,[7] opencv,[8]

[6]https://pytorch.org/
[7]https://numpy.org/
[8]https://opencv.org/

scikit-image,[9] plotly,[10] pandas,[11] matpoltlib[12] and scikit-learn.[13] Furthermore, existing predictors are not evaluated on same benchmark datasets. To evaluate the performance of proposed predictors, authors of existing studies [3], [5], [13], [24], [50], [66], [70] utilize simulators to generate their datasets which are not publicly available. The source codes of only 2 predictors (Shvejan et al. [66], Bayarov et al. [1]) are publicly available. To implement other 6 predictors, we follow methodologies, hyper-parameters and experimental settings details from their research articles [3], [5], [13], [24], [50], [70]. In order to evaluate the effectiveness of proposed RPRP-SAP predictor we take 8 most recent existing predictors as baseline methods for comparative performance analysis.

TABLE 3. Search space and optimal values of different hyper-parameters for proposed predictor.

Hyper-parameter	Description	Search Space	Optimal Value
Learning Rate	A hyper-parameter that controls model's response to error feedback	0.000001 to 0.1	0.0001
Weight Decay	To reduce the complexity of the model and enable generalizability of model	0.00001 to 0.3	0.0001
Dropout	To drop data features randomly in order to avoid over-fitting of model	0.1 to 0.7	0.2
ResNet Block	Stack of layers to extract informative patterns of data	1 to 5	4

It is widely accepted that hyper-parameters influence the performance of deep learning predictors [64]. We perform large-scale experimentation to find the best values of hyper-parameters from a pool of predefined search space corresponding to each hyper-parameter. Specifically, we optimize 3 hyper-parameters: learning rate [2], weight decay [75] and dropout [22]. Proposed RPRP-SAP predictor architecture is made up of ResNet blocks, so it is important to select an appropriate number of modules [31]. To find optimal ResNet blocks, we perform experimentation by a varying number of blocks from 1 to 5. The search space of ResNet blocks and hyper-parameters along with a brief description and selected optimal values are provided in Table 3.

V. RESULTS

This section provides a detailed performance comparison of proposed and 8 existing predictors [1], [3], [5], [13], [24], [50], [66], [70] on two real-world and two simulated datasets. It first illustrates a comprehensive performance comparison of proposed and existing predictors in same-track evaluation setting. Then it compares their performance in cross-track evaluation setting where predictors are trained on one track data and are evaluated on other track data. It also illustrates performance of predictors in cross-domain evaluation setting where they are trained on simulated datasets and are evaluated on real-world datasets and vice versa.

A. PERFORMANCE COMPARISON OF PROPOSED AND EXISTING PREDICTORS UNDER SAME-TRACK EVALUATION SETTING

Table 4 compares the performance of proposed and 8 existing predictors [1], [3], [5], [13], [24], [50], [66], [70] under same-track evaluation setting using 5 different datasets. It is evident from Table 4 over lake track dataset, among 6 existing CNN-based predictors [1], [3], [5], [13], [24], [50] Mohammadi et al. predictor [50] produces the highest performance while Anchalia et al. predictor [5] produces least performance. Primarily, Mohammadi et al. predictor [50] produces better performance because it utilizes features that are manually crafted through traditional computer vision strategies to find regions of interest (ROI) from input images and aid to eliminate irrelevant and redundant regions. The extracted ROIs of images are fed to a deeper CNN architecture which extracts informative features and utilizes them for finding accurate prediction of steering angle. Anchalia et al. [5] predictor makes use of shallow CNN architecture hence, probably fails to extract significant features and leads predictor toward worst performance. Garg et al. [24] predictor manages to produce the 2^{nd} highest performance followed by Kalim et al. [13] predictor. Furthermore, Ali et al. [3] and Bayarov et al. [1] predictors get 4^{th} and 5^{th} ranks, respectively. These predictors fail to produce better performance because these approaches are using deep networks that hinder proper flow of gradients.

On the other hand, Valiente et al. [70] hybrid predictor produces almost similar performance to CNN-based top-performing predictor [50]. Although, unlike Mohammadi et al. [50] predictor, Valiente et al. [70] predictor does not make use of extra features extracted through traditional computer vision strategies but it reaps the benefits of both CNN and LSTM architectures to preserve temporal information. Shvejan et al. [66] predictor that employed advance transformer unit [26], only manages to produce performance better than 3 least performing CNN-based predictors [3], [5], [66]. Primarily, the designed architecture is very deep and complex as it utilizes eight stacks of transformer architecture [26]. Consequently, extracted features are deprived of high-level representation and can only access representations at previous layers, resulting in poor performance. Proposed RPRP-SAP predictor produces better performance than all existing predictors across both evaluation measures. Although proposed RPRP-SAP predictor also relies upon convolutional layers yet it is competent in extracting more comprehensive features due to proper

[9] https://scikit-image.org/
[10] https://plotly.com/
[11] https://pandas.pydata.org/
[12] https://matplotlib.org/
[13] https://scikit-learn.org/stable/

TABLE 4. Performance comparison of proposed and existing predictors over lake track, jungle track, CARLA, Sully Chen and comma.ai datasets under same-track evaluation setting.

Predictor	Trainable Parameters	Lake Track		Jungle Track		CARLA		Sully Chen		Comma.ai	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Ali et al. [3]	252,225	0.41	0.50	0.28	0.35	0.093	0.178	16.88	30.68	0.527	0.706
Garg et al. [24]	252,219	0.21	0.26	0.32	0.41	0.071	**0.176**	11.74	24.67	0.593	0.798
Anchalia et al. [5]	3,921	1.53	1.67	0.33	0.41	0.267	0.316	15.63	27.44	0.598	0.852
Shvejan et al. [66]	4,638,465	0.31	0.36	0.31	0.39	0.071	**0.176**	13.02	25.40	0.523	0.816
Valiente et al. [70]	288,599	0.20	0.25	0.28	0.35	0.073	**0.176**	12.47	25.04	0.510	0.789
Bayarov et al. [1]	252,225	0.42	0.50	0.27	0.33	0.126	0.218	11.94	24.73	0.691	0.985
Kalim et al. [13]	437,697	0.25	0.31	0.35	0.43	0.073	**0.176**	12.75	25.35	**0.491**	0.78
Mohammadi et al. [50]	252,225	0.20	0.254	0.27	0.34	0.073	31.432	**0.176**	24.91	0.896	0.644
Proposed	2,816,225	**0.13**	**0.13**	**0.19**	**0.30**	**0.065**	**0.176**	2.41	**8.80**	0.500	**0.506**

gradient flow while learning weights. In comparison to existing predictors, a unique feature of proposed RPRP-SAP predictor is residual unit [31] which provides alternative paths for proper flow of gradient during back-propagation.

Jungle track data is relatively more complex and challenging than lake track in terms of feature extraction. Among 6 CNN-based predictors [1], [3], [5], [13], [24], [50] Mohammadi et al. [50] and Bayarov et al. [1] predictors produce nearly similar and best performance over jungle track data. The hand-crafted feature based predictor [50] applies convolution operation to only important regions of images while decision of other predictor [1] is based on the combination of convolutional and dense layers. The higher performance of these predictors is due to their remarkable potential to extract comprehensive feature patterns from complex occluded environments of jungle track.

Moreover, the remaining 4 CNN-based predictors including; Ali et al. [3], Garg et al. [24], Anchalia et al. [5] and Kalim et al. [13] predictors are ranked at positions 3, 4, 5 and 6 in terms of performance. Although the performance of Valiente et al. [70] hybrid predictor is lower than 2 top-performing CNN predictors [1], [50] but it manages to beat performance of rest of the CNN-based predictors. Shvejan et al. [66] predictor once again failed to perform on this dataset and could only beat performance of 3 least performing CNN-based predictors [5], [13], [24]. Table 4 shows that the proposed predictor once again outperforms all existing predictors for both evaluation measures on jungle track dataset.

Among existing predictors evaluated on CARLA dataset, Shvejan et al. predictor [66] secures 1^{st} rank and showcases the adaptability of vision transformers in capturing complex visual patterns. Garg et al. predictor [24] secures 2^{nd} ranks by extracting most informative features. The Valiente et al. predictor [70] secures 3^{rd} rank by fusing both spatial and temporal features. Mohammadi et al. predictor [50] illustrates the potential of blending traditional hand-crafted features with modern neural networks. This hybrid approach leverages domain-specific knowledge and secures 4^{th} position. Kalim et al. [13], Ali et al. [3] and Bayarov et al. [1] predictors utilizes convolution features to capture diverse information and secures 5^{th}, 6^{th} and 7^{th} positions, respectively. Anchalia et al. [5] shallow predictor fails to extract meaningful information and remains worst performer among all existing predictors. The proposed predictor stands out as the top-performing, demonstrating the lowest MAE.

A thorough analysis of Table 4 reveals consistent trend across three simulated datasets (lake track, jungle track and CARLA) regarding the performance of various existing predictors. Specifically, Ali et al. [3] Bayarov et al. [1] and Anchalia et al. [5] consistently exhibit lowest predictive performance among all existing predictors across these datasets. Conversely, the remaining 5 existing predictors [13], [24], [50], [66], [70] demonstrate performance variability across different datasets however, their MAE error values generally fall within a similar range.. Additionally, all existing predictors demonstrate superior performance when tested on the CARLA dataset, in contrast to their performance on 2 versions of Udacity dataset. Particularly, worst performing predictor [5] over CARLA dataset produces MAE error approximately equal to best performing predictors [13], [24], [50], [70] over jungle and lake tracks. This significant performance difference demonstrates that CARLA simulator offers a more informative features for accurate steering angle prediction.

It is evident from Table 4 all existing predictors produce higher MAE and RMSE error values on Sully Chen data in comparison to their error values on simulated datasets. Among existing predictors, Garg et al. [24] predictor manages to produce best performance and Ali et al. [3] predictor produces least performance over Sully Chen dataset [43]. Least performing predictor [5] of lake track dataset also failed to perform well on Sully Chen dataset [43] and secures 2^{nd} last rank in terms of MAE and RMSE errors among existing predictors. In contrast to this, 2 best performing predictors [50], [70] with same MAE error over lake track data secure 3^{rd} and 6^{th} rank on Sully Chen dataset, respectively.

Although top performing predictor [1] over jungle track dataset secures 2^{nd} rank over Sully Chen data yet it produces high MAE and RMSE errors of 11.94 and 24.73, respectively. Furthermore, worst performing predictor [13] over jungle track data secures 3^{rd} position on Sully Chen dataset with 12.75 MAE and 25.75 RMSE. Hence, all existing predictors failed to capture generalized feature patterns and show highly fluctuating performance over all 3 datasets. However, proposed predictor once again remarkably beats the performance of all existing predictors with MAE and RMSE errors of 2.41 and 8.80, respectively. A fair performance analysis of proposed and existing predictors in terms of MAE and RMSE over simulated and real-world dataset indicates that proposed predictor is capable of effectively generalizing over diverse types of dataset by learning comprehensive and informative patterns of data.

Table 4 demonstrates a north-worthy performance contrast of existing predictors over two distinct real-world datasets. All existing predictors show better performance over comma.ai dataset with lower MAE errors in comparison to their performance over Sully Chen dataset. Specifically, top-performing predictor Garg et al. [24] over Sully Chen dataset, secured 5^{th} rank over comma.ai dataset. Conversely, the least performing predictor [3] over Sully Chen dataset, secured 4^{th} rank over comma.ai dataset. A similar pattern can be observed across remaining predictors. This significant MAE error difference across two real-world datasets highlights diverse data characteristics and challenges that impact predictors performance.

1) PERFORMANCE COMPARISON OF PROPOSED AND EXISTING PREDICTORS BASED ON AUROC

This section summarizes performance comparison of proposed RPRP-SAP predictor with 8 existing predictors using Area Under Receiver Operating Characteristics (AUROC) under same track evaluation settings over 5 distinct datasets. In order to generate AUROC scores of predictors, the actual and predicted values of steering angle are divided into 3 bins consisting of right, left and straight directions.

Critical analysis of AUROC graphs shown in Figure 5 for simulated and real datasets illustrates that proposed predictor attains the highest performance in comparison to all 8 existing predictors. It is evident from Figure 5 that Mohammadi et al. [50] predictor performs nearly equal to proposed predictor for jungle track and lake tracks but achieves last, 2^{nd} last and 3^{rd} last rank on comma.ai, Sully Chen and CARLA datasets, respectively. However, Anchalia et al. [5], Shvejan et al. [66] and Valiente et al. [70] predictors show poor performance across simulated datasets but attain decent and nearly equal performance score over real dataset. These 2 predictors [5], [66] exhibit a similar performance trend across CARLA dataset, but Valiente et al. predictor [70] lags behind in performance. Bayarov et al. [1] predictor performs poorly on jungle track, CARLA and Sully Chen data, even though it is the second best performing method on lake track and Sully Chen datasets. In contrast to this Ali et al. [3] and Kalim et al. [13] predictors show remarkable performance on jungle track, CARLA and comma.ai dataset and achieve relatively lower performance on lake track and Sully Chen data.

Garg et al. [24] predictor performs well on jungle and lake tracks and fails to manage good performance over CARLA, Sully Chen and comma.ai datasets. By comparing the performance values of the proposed RPRP-SAP predictor with 8 existing predictors, we can conclude that the proposed predictor is capable of comprehending the significant features of diverse environments across all 5 datasets for steering angle prediction of AVs.

2) COMPARISON OF PROPOSED AND EXISTING PREDICTORS IN TERMS OF ACTUAL VS PREDICTED STEERING ANGLE DIRECTION

In order to facilitate deep insight into predictor's behavior for estimating steering angle of AVs on road track, this section provides a graphical analysis of actual and predicted steering direction computed under same-track evaluation setting, over 5 datasets. For this analysis, actual and predicted steering angles are divided into 3 bins based on numerical values of steering angle corresponding to 3 directions namely; right, left and straight direction.

It is evident from Figure 6 that over lake and jungle track datasets Anchalia et al. [5] and Shvejan et al. [66] predictors are worst performers as both predictors only predict left steering direction within a range of -0.5 to -0.1 and hamper the prediction of straight or right direction. However, for jungle track dataset both predictors display a higher biasness towards left most direction with a steering angle of -1. Conversely, over CARLA dataset, Anchalia et al. predictor [5] exhibits consistent behavior, while Shvejan et al. predictor [66] effectively predicts straight, right and left steering angles. Similar to these predictors Mohammadi et al. [50] and Kalim et al. [13] predictors are more biased toward left direction and do not predict straight or right direction over jungle and lake tracks datasets. However, over CARLA dataset, both predictors [13], [50] exhibit a significant bias towards

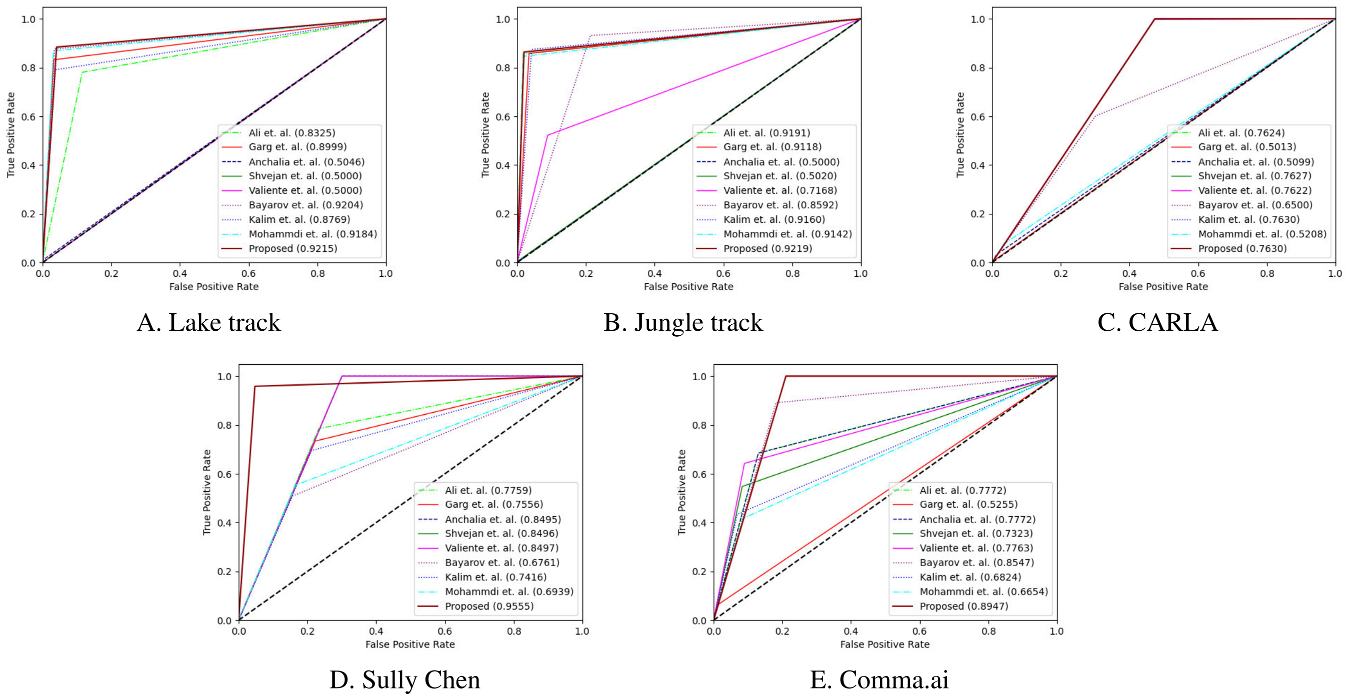


FIGURE 5. **AUROC of simulated and real-world datasets under same-track evaluation setting.**

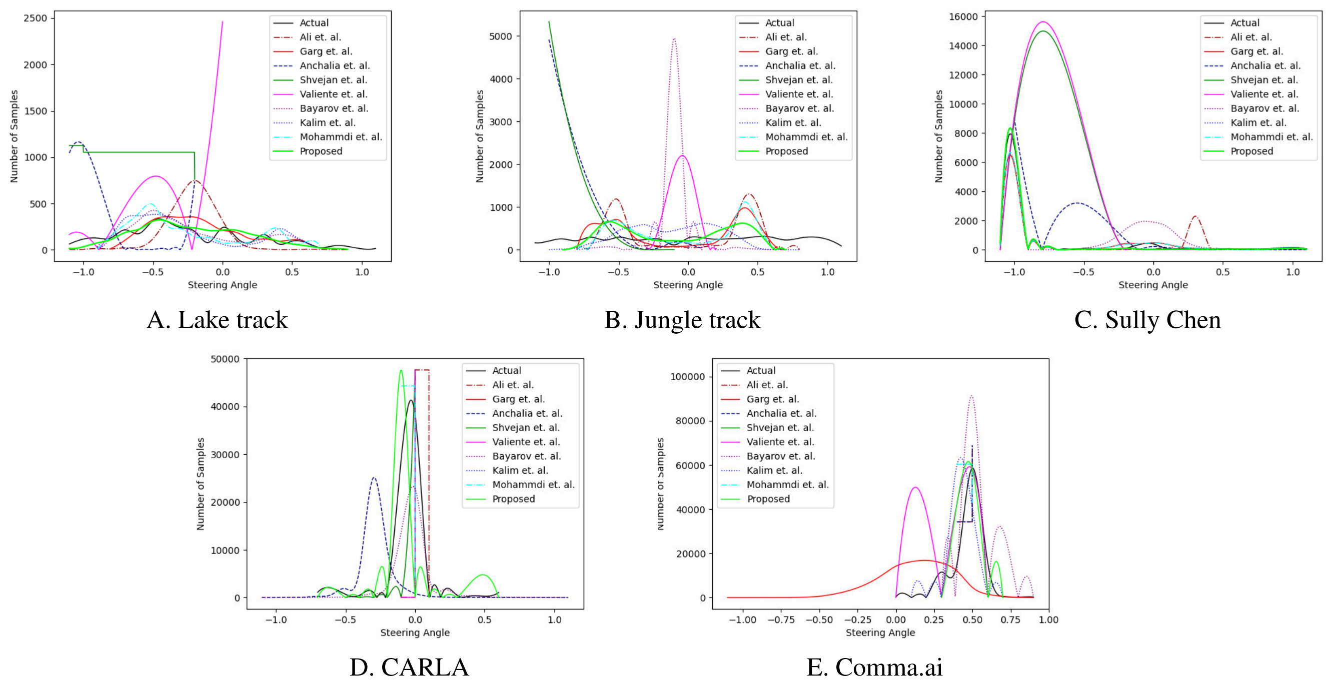


FIGURE 6. **Actual vs predicted steering angles of proposed and existing predictors over simulated and real-world datasets under same-track evaluation setting.**

predicting straight steering angles, with limited predictions for right or left steering angles.

However, Garg et al. [24] Valiente et al. [70] predictors show an equal balance of estimated straight, right and left steering directions across all 3 simulated datasets. Bayarov et al. [1] predictor shows least standard deviation [48] and the highest peak at straight direction over all 3 simulated datasets. Ali et al. predictor [3] exhibits a highly unusual behavior and consistently predicts either left or right steering directions over jungle track while predominantly predicts straight or right steering angles over CARLA dataset. However, predicted angle varies between left, right and straight directions over lake track but shows a lower standard deviation to right direction in comparison to left direction.

Over, Sully Chen dataset, existing predictors exhibit very poor performance in terms of predicted directions. Most of the existing predictors [5], [13], [24], [50], [66], [70] are highly biased and always predict left steering direction. Moreover, few existing predictors [1], [3] show deviations from the actual steering angles in all 3 directions. The critical analysis of Figure 6 illustrates that proposed predictor beats performance of existing predictor over all 3 datasets and more precisely predicts the steering angle along all directions with a greater standard deviation which is nearly similar to standard deviation of actual values of steering direction. Across comma.ai dataset, Valiente et al. [70], Kalim et al. [13], Shvejan et al. [66] and Ali et al. [3] demonstrate effective predictions of diverse steering angles, while a few predictors (Anchalia et al. [5], Bayarov et al. [1] and Mohammadi et al. [50]) consistently predict straight steering angles. In contrast, Garg et al. predictor [24] displays an even distribution of all three steering angle categories across all data samples. Furthermore, Figure 6 illustrates that proposed RPRP-SAP predictor is competent to effectively predict a wide range of steering angles across all five distinct datasets and showcases its ability to generalize and adapt to various environmental conditions for autonomous driving.

B. PERFORMANCE COMPARISON OF PROPOSED AND EXISTING PREDICTORS IN CROSS-TRACK EVALUATION SETTING

This section compares performance of proposed RPRP-SAP and 8 existing predictors [1], [3], [5], [13], [24], [50], [66], [70] under cross-track evaluation setting. Mainly, this analysis explores robustness of predictors during real-world deployment because usually AVs are tested on multiple road tracks.

Figure 7 illustrates cross-track setting based performance of proposed and 8 existing predictors [1], [3], [5], [13], [24], [50], [66], [70] in terms of 2 different evaluation measures across Udacity dataset. Among 6 existing CNN-based predictors [1], [3], [5], [13], [24], [50] Garg et al. [24] predictor manages to produce the highest and Kalim et al. [13] predictor achieves 2^{nd} the highest performance for both tracks. Although under same-track evaluation setting, transformer-based Shvejan et al. [66] predictor remained 2^{nd} least performer due to its complex architecture, hence because of similar drawback, it once again fails to produce decent performance over both tracks under cross-track evaluation setting.

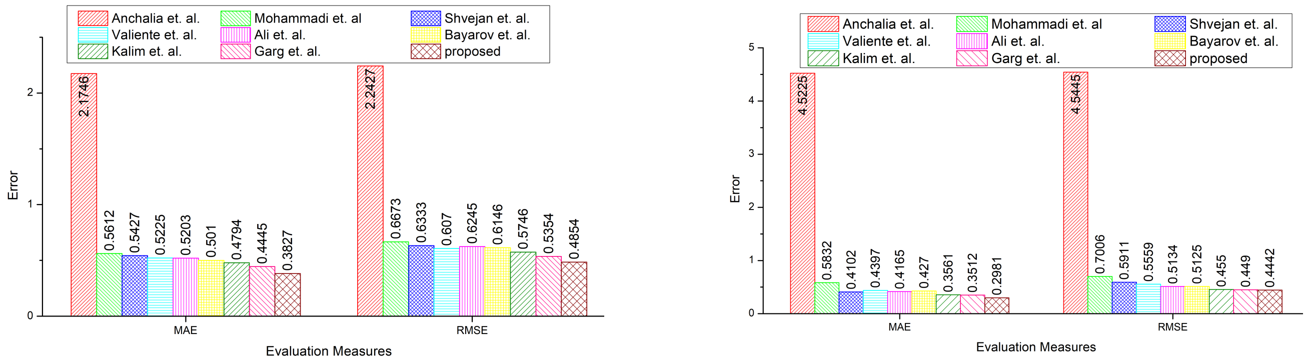
Similar to transformer based predictor [66], hybrid predictor [70] also fails to retain its rank achieved under same-track evaluation setting, for this setting it achieved 1^{st} rank over lake track data while under cross-track evaluation setting over dataset of same-track data it achieves 5^{th} rank. Similarly, for jungle track dataset, under same-track evaluation setting it manages to achieve 4^{th} ranked performance while in cross-track evaluation setting its performance further decreased from 3 predictors including Garg et al. [24] Kalim et al. [13] and Shvejan et al. [66] predictors which had a lower performance than hybrid predictor under same-track evaluation setting. It is considered that hand-crafted features based predictor fails to produce a similar performance on different data types of same task, because if a traditional feature extraction approach extracts comprehensive features from one particular data it may not extract similar types of features from slightly different data for same task due to background noise or difference of important feature patterns between 2 datasets. Hand-crafted features based predictor [50] manages to produce the best performance under same-track evaluation setting for both tracks but it completely fails to produce similar performance in cross-track evaluation setting and became second last performer among 8 different predictors.

A critical performance analysis of 8 existing predictors, in 2 different types of evaluation settings, demonstrates that simplest and most complex architectures fails to produce better performance, as the predictor developed by Anchalia et al. [5] is very simple and it remains worst performer under both experimental settings for both datasets.

Furthermore, we also evaluated performance of proposed RPRP-SAP and 8 existing predictors under cross-track evaluation setting using CARLA dataset. Specifically, we used 5 distinct experimental settings, where one town is separated for testing while the remaining towns are collectively used for predictors training. For instance in first setting, town 1 is used for evaluation, while town 2, 3, 4 and 5 are used for training. This experimental setting is repeated for all 5 towns. Table 5 illustrates performance comparison of existing and proposed predictors under 5 different experimental settings over CARLA dataset.

It is evident from Table 5, among 6 CNN based predictors [1], [3], [5], [13], [24], [50] 3 predictors [1], [24], [50] achieve relatively low MAE and RMSE values across all 5 experimental settings, indicating good predictive accuracy. The better performance of these 3 CNN based predictors [1], [24], [50] is due to incorporation of enhanced features extraction [50], elimination of irrelevant features [24] and improved features distribution [1]. The remaining 3 CNN based predictors [3], [5], [13] failed to capture complex features due to simple architectures and generally had higher MAE and RMSE values across all 5 experimental settings. Particularly, Anchalia et al. predictor [5] fails to make accurate predictions on towns 3 and 4 while Kalim et al. predictor [13] yields relatively higher MAE and RMSE values on town 1. On the other hand, Ali et al. predictor [3] demonstrates slightly higher MAE and RMSE values on towns 4 and 5.

Valiente et al. hybrid predictor [70] combines strength of CNN and LSTM to capture spatial and temporal features and



A. Lake track

B. Jungle track

FIGURE 7. Performance comparison of proposed and existing predictors in terms of MAE and RMSE over lake track and jungle track dataset under cross-track evaluation setting.

TABLE 5. Performance comparison of proposed and existing predictors in terms of MAE and RMSE over CARLA dataset under cross-track evaluation settings.

Experimental Setting	Training Data	Town 2 3 4 5		Town 1 3 4 5		Town 1 2 4 5		Town 1 2 3 5		Town 1 2 3 4	
	Validation Data	Town1		Town 2		Town 3		Town 4		Town 5	
Predictor	Trainable Parameters	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Ali et al. [3]	252,225	0.074	0.163	0.085	0.199	0.076	**0.168**	0.086	0.173	0.078	0.178
Garg et al. [24]	252,219	0.056	0.163	0.087	0.199	0.070	**0.168**	0.065	0.171	0.068	0.177
Anchalia et al. [5]	3,921	0.066	0.163	0.117	0.206	0.074	0.169	0.070	**0.170**	0.077	0.177
Shvejan et al. [66]	4,638,465	0.059	**0.162**	0.091	0.199	0.072	**0.168**	0.062	0.171	0.072	0.177
Valiente et al. [70]	288,599	0.057	0.163	0.087	0.199	0.070	**0.168**	0.066	0.171	0.069	0.177
Bayarov et al. [1]	252,225	0.058	0.163	0.082	0.199	0.067	**0.168**	0.063	0.171	0.069	0.177
Kalim et al. [13]	437,697	0.076	0.179	0.081	**0.169**	0.070	0.177	0.069	13.14	0.068	**0.175**
Mohammadi et al. [50]	252,225	0.060	0.163	0.084	0.199	0.074	0.169	0.063	0.171	0.068	0.177
Proposed	2,816,225	**0.055**	0.163	**0.079**	0.199	**0.064**	**0.168**	**0.061**	0.171	**0.067**	0.177

consistently performed well on all towns expect town 2 where it shows slightly higher errors. Moreover, transformer based predictor [66] successfully extracts complex features and achieves relatively consistent performance with minor fluctuations in MAE and RMSE values across all 5 experimental settings except town 2, where it shows slightly higher errors.

A thorough performance analysis of existing predictors [1], [3], [5], [13], [24], [50], [66], [70] reveals that overall 5 predictors [1], [24], [50], [66], [70] demonstrates good generalization potential across all 5 experimental settings. However, it can be seen in Table 5 that proposed RPRP-SA predictor consistently outperforms existing predictors under all 5 experimental settings with lowest MAE and RMSE errors. This consistent outstanding performance of proposed RPRP-SA predictor across diverse environmental settings highlights its robustness and effectiveness.

C. PERFORMANCE COMPARISON OF PROPOSED AND EXISTING PREDICTORS UNDER CROSS-DOMAIN EVALUATION SETTING

Table 6 illustrates performance comparison of proposed and existing predictors across 2 real-world and 2 simulated datasets under cross-domain evaluation setting. In this experimental setup, each predictor undergoes training on every real-world dataset and is subsequently evaluated using

TABLE 6. Performance comparison of proposed and existing predictors in cross-domain evaluation setting over simulated and real-world datasets.

Experimental Setting	Training Data	Udacity Simulator				CARLA Simulator				Sully Chen				Comma. ai			
	Evaluation Data	Comma. ai		Sully Chen		Comma. ai		Sully Chen		Udacity Simulator		CARLA Simulator		Udacity Simulator		CARLA Simulator	
Evaluation Measure		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Predictors	Bayarov et al. [1]	0.503	**0.511**	15.01	31.424	5.976	6.329	14.96	31.401	18.082	19.612	66.226	66.226	4.692	4.741	0.661	0.683
	Valiente et al. [70]	0.527	0.524	14.987	31.413	0.505	0.512	15.029	31.434	1.758	**0.207**	1.72	1.729	0.648	0.719	0.502	0.522
	Kalim et al. [13]	0.583	0.589	14.88	31.366	0.501	0.508	14.869	31.354	0.749	2	8.807	11.569	10.663	11.1	0.508	0.528
	Anchalia et al. [5]	4.112	4.113	14.959	31.394	0.499	**0.503**	15.008	31.417	12.378	15.064	16.564	16.565	3.81	4.588	0.12	0.186
	Mohammadi et al. [50]	0.555	0.561	15.036	31.419	0.502	0.504	15.095	31.448	14.648	22.284	3.371	3.375	**0.511**	0.693	0.097	0.186
	Garg et al. [24]	0.533	0.539	14.894	31.376	0.504	0.511	14.904	31.377	1.812	2.183	1.184	1.198	1.459	1.515	0.504	0.524
	Ali et al. [3]	0.501	0.517	15.017	31.426	0.506	0.512	15.064	31.435	18.418	25.711	23.039	23.039	5.839	5.897	0.528	0.549
	Shvejan et al. [66]	7.015	7.016	15.037	31.43	0.504	0.51	15.062	31.449	13.593	18.536	11.114	11.116	0.597	**0.67**	0.518	0.538
	Proposed	**0.497**	0.599	**12.83**	**25.031**	**0.497**	**0.503**	**12.83**	**25.031**	**0.65**	0.799	**0.501**	**0.523**	8.767	22.805	**0.093**	**0.18**

each simulation-based dataset. Likewise, each predictor is trained on each simulation-based dataset and assessed with each real-world dataset.

A critical analysis of Table 6 reveals that, when the predictors are trained using data from the Udacity simulator and then evaluated on the real data from comma.ai, they exhibit improved performance compared to when they are trained on the same dataset but evaluated on the Sully Chen dataset. A similar trend in performance is observed when the predictors are trained on the CARLA dataset and then evaluated on both the Sully Chen and comma.ai datasets.

Conversely, when predictors are trained on the Sully Chen dataset and then evaluated on both simulated datasets, most of the predictors produce a higher error rate compared to their produced error when they are trained on the comma.ai dataset and evaluated on the same simulated datasets. Only two predictors (proposed and Kalim et al. [13]) produce higher errors when trained on comma.ai dataset and evaluated on Udacity dataset. Overall, performance analysis suggests that the comma.ai dataset closely resembles the characteristics of the simulated datasets (Udacity and CARLA), while the graphical features of the Sully Chen dataset differs from those of the simulation-based datasets.

Moreover, when the predictors are trained using the Udacity dataset and evaluated on the comma.ai dataset, it is observed that out of the 9 predictors, 6 [1], [3], [13], [24], [66], [70] of them exhibit better performance compared to when they are trained on the comma.ai dataset and evaluated on the Udacity dataset. Similarly, when these predictors are trained on the CARLA dataset and evaluated on the comma.ai dataset, 4 [3], [13], [24], [66] out of the 9 predictors show better performance compared to when they are trained on the comma.ai dataset and evaluated on the CARLA dataset.

Conversely, for most of the predictors [1], [5], [13], [24], [50], [70], their performance is better when trained on real-world data from Sully Chen and evaluated on simulated datasets, as opposed to when they are trained on simulated datasets and evaluated on Sully Chen's real-world data. In all eight different scenarios, it is observed that in most of the cases the proposed predictor outperforms existing predictors in terms of performance. In summary, it can be concluded that the predictors tend to perform better when they are trained on simulator data from Udacity and CARLA and then evaluated on real data from comma.ai.

VI. CONCLUSION

Autonomous vehicles are actively contributing to avoid haphazard situations and alleviate the risk of accidents particularly caused by carelessness of drivers. However, designing and testing complex intelligent systems of autonomous vehicles is time-consuming and costly. To cope with this limitation, several simulation environments are developed for efficient designing of AI-supported intelligent systems and to perform their testing. To empower the process of steering angle prediction, this study presents different versions of two benchmark datasets that are generated through Udacity and CARLA simulators. It presents a ResNet based predictor and benchmarks the performance of proposed and existing predictors.

It explores the potential of proposed and existing predictors under 3 different experimental settings i.e. same-track, cross-track and cross-domain. Under same track evaluation setting where predictors are trained and evaluated across real world and simulators same track datasets, experimental results reveal under this evaluation setting predictors produce better performance on simulated and real-world datasets. Contrarily, under cross track evaluation setting, where predictors are trained on simulated datasets of one track and are evaluated on other track, experimental results reveals difference between training and testing scenarios pose obstacles in generalizing to complex driving conditions resulting in poor performance. Furthermore, experimental results under cross domain setting, where predictors are trained on simulated dataset and are evaluated on real world dataset and vice versa, experimental results reveal that simulators generated data is closer to comma.ai real-world data but seems different from Sully Chen data. These findings emphasize the need to meticulously consider the diversity and complexity of training datasets to ensure robust performance across wide range of real-world driving scenarios.

In this study, our primary focus is on investigating the potential of various predictors that are based on simulators. However, as we move forward, there is a compelling direction for further research. This direction entails selecting a subset of the most effective simulator-based predictors, alongside a selection of predictors that rely on real-world data. We aim to assess their capabilities in handling diverse real-world datasets and evaluate their performance in a cross-domain setting.

ACKNOWLEDGMENT

(Summra Saleem and Muhammad Nabeel Asim contributed equally to this work.)

REFERENCES

[1] H. B. Ahmedov, D. Yi, and J. Sui, "Application of a brain-inspired deep imitation learning algorithm in autonomous driving," *Softw. Impacts*, vol. 10, Nov. 2021, Art. no. 100165.
[2] R. A. Al-Jawfi, "The effect of learning rate on fractal image coding using artificial neural networks," *Fractal Fractional*, vol. 6, no. 5, p. 280, May 2022.
[3] S. Ali, "Self-driving cars: Automation testing using Udacity simulator," *Int. Res. J. Eng. Technol.*, vol. 8, no. 4, pp. 1–7, 2021.
[4] M. Anand, A. Velu, and P. Whig, "Prediction of loan behaviour with machine learning models for secure banking," *J. Comput. Sci. Eng.*, vol. 3, no. 1, pp. 1–13, Feb. 2022.
[5] J. Anchalia and S. Srividhya, "Building CNN model for autonomous car using Udacity simulator," *Recent Trends Comput. Graph. Multimedia Technol.*, vol. 4, no. 1, 2022.
[6] K. L.-M. Ang, J. K. P. Seng, E. Ngharamike, and G. K. Ijemaru, "Emerging technologies for smart cities' transportation: Geo-information, data analytics and machine learning approaches," *ISPRS Int. J. Geo-Inf.*, vol. 11, no. 2, p. 85, Jan. 2022.
[7] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.
[8] L. Bergamini, Y. Ye, O. Scheel, L. Chen, C. Hu, L. D. Pero, B. Osinski, H. Grimmett, and P. Ondruska, "SimNet: Learning reactive self-driving simulations from real-world observations," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 5119–5125.
[9] A. S. Bhadoriya, V. K. Vegamoor, and S. Rathinam, "Object detection and tracking for autonomous vehicles in adverse weather conditions," SAE Tech. Paper 2021-01-0079, 2021.
[10] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Müller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," 2016, *arXiv:1604.07316*.
[11] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Proc. 11th Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2010, pp. 778–792.
[12] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Comput. Sci.*, vol. 7, p. e623, Jul. 2021.
[13] M. K. A. Chy, A. K. M. Masum, K. A. M. Sayeed, and M. Z. Uddin, "Delicar: A smart deep learning based self driving product delivery car in perspective of Bangladesh," *Sensors*, vol. 22, no. 1, p. 126, Dec. 2021.
[14] *The comma.ai Driving Dataset*. Accessed: Oct. 2, 2023. [Online]. Available: https://github.com/commaai/research
[15] R. D. O. Werneck, R. Prates, R. Moura, M. M. Gonçalves, M. Castro, A. Soriano-Vargas, P. Ribeiro Mendes Júnior, M. M. Hossain, M. F. Zampieri, A. Ferreira, A. Davólio, D. Schiozer, and A. Rocha, "Data-driven deep-learning forecasting for oil production and pressure," *J. Petroleum Sci. Eng.*, vol. 210, Mar. 2022, Art. no. 109937.
[16] H. Defienne, P. Cameron, B. Ndagano, A. Lyons, M. Reichert, J. Zhao, A. R. Harvey, E. Charbon, J. W. Fleischer, and D. Faccio, "Pixel super-resolution with spatially entangled photons," *Nature Commun.*, vol. 13, no. 1, p. 3566, Jun. 2022.
[17] O. Doguc, "Robot process automation (RPA) and its future," in *Research Anthology on Cross-Disciplinary Designs and Applications of Automation*. Hershey, PA, USA: IGI Global, 2022, pp. 35–58.
[18] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. Conf. Robot Learn.*, 2017, pp. 1–16.
[19] S. Du, H. Guo, and A. Simpson, "Self-driving car steering angle prediction based on image recognition," 2019, *arXiv:1912.05440*.
[20] H. Etienne, "The dark side of the 'moral machine' and the fallacy of computational ethical decision-making for autonomous vehicles," *Law, Innov. Technol.*, vol. 13, no. 1, pp. 85–107, Jan. 2021.
[21] D. J. Fremont, E. Kim, Y. V. Pant, S. A. Seshia, A. Acharya, X. Bruso, P. Wells, S. Lemke, Q. Lu, and S. Mehta, "Formal scenario-based testing of autonomous vehicles: From simulation to the real world," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–8.
[22] C. Garbin, X. Zhu, and O. Marques, "Dropout vs. batch normalization: An empirical study of their impact to deep learning," *Multimedia Tools Appl.*, vol. 79, nos. 19–20, pp. 12777–12815, May 2020.
[23] M. Gardner et al., "Evaluating models' local decision boundaries via contrast sets," 2020, *arXiv:2004.02709*.
[24] N. Garg, K. S. Ashrith, G. S. Parveen, K. G. Sai, A. Chintamaneni, and F. Hasan, "Self-driving car to drive autonomously using image processing and deep learning," *Int. J. Res. Eng., Sci. Manage.*, vol. 5, no. 1, pp. 125–132, 2022.
[25] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, in JMLR Workshop and Conference Proceedings, 2011, pp. 315–323.
[26] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 15908–15919.
[27] L. Han, L. Wu, F. Liang, H. Cao, D. Luo, Z. Zhang, and Z. Hua, "A novel end-to-end model for steering behavior prediction of autonomous ego-vehicles using spatial and temporal attention mechanism," *Neurocomputing*, vol. 490, pp. 295–311, Jun. 2022.
[28] F. U. Haq, D. Shin, S. Nejati, and L. C. Briand, "Comparing offline and online testing of deep neural networks: An autonomous car case study," in *Proc. IEEE 13th Int. Conf. Softw. Test., Validation Verification (ICST)*, Oct. 2020, pp. 85–95.
[29] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, 1988, pp. 10–5244.
[30] I. U. Hassan, H. Zia, H. S. Fatima, S. A. Yusuf, and M. Khurram, "A lightweight convolutional neural network to predict steering angle for autonomous driving using CARLA simulator," *Model. Simul. Eng.*, vol. 2022, pp. 1–11, Aug. 2022.
[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[32] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands. Cham, Switzerland: Springer, Oct. 2016, pp. 630–645.
[33] G. Hou, Z. Pan, B. Huang, G. Wang, and X. Luan, "Hue preserving-based approach for underwater colour image enhancement," *IET Image Process.*, vol. 12, no. 2, pp. 292–298, Feb. 2018.
[34] Y. Hou, Z. Ma, C. Liu, and. C. Loy, "Learning to steer by mimicking features from heterogeneous auxiliary networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8433–8440.
[35] N. Ijaz and Y. Wang, "Automatic steering angle and direction prediction for autonomous driving using deep learning," in *Proc. Int. Symp. Comput. Sci. Intell. Controls (ISCSIC)*, Nov. 2021, pp. 280–283.
[36] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
[37] A. Kalatian and B. Farooq, "A context-aware pedestrian trajectory prediction framework for automated vehicles," *Transp. Res. C, Emerg. Technol.*, vol. 134, Jan. 2022, Art. no. 103453.
[38] D. S. K. Karunasingha, "Root mean square error or mean absolute error? Use their ratio as well," *Inf. Sci.*, vol. 585, pp. 609–629, Mar. 2022.

[39] D. Khaledyan, A. Amirany, K. Jafari, M. H. Moaiyeri, A. Z. Khuzani, and N. Mashhadi, "Low-cost implementation of bilinear and bicubic image interpolation for real-time image super-resolution," in *Proc. IEEE Global Humanitarian Technol. Conf. (GHTC)*, Oct. 2020, pp. 1–5.

[40] Q. Khan, T. Schön, and P. Wenzel, "Latent space reinforcement learning for steering angle prediction," 2019, *arXiv:1902.03765*.

[41] W. Ko, S. Park, S. Park, H. Jeong, and I. Yun, "Development of freeway-based test scenarios for applying new car assessment program to automated vehicles," *PLoS ONE*, vol. 17, no. 7, Jul. 2022, Art. no. e0271532.

[42] M. T. R. Laskar, E. Hoque, and J. X. Huang, "Domain adaptation with pre-trained transformers for query-focused abstractive text summarization," *Comput. Linguistics*, vol. 48, no. 2, pp. 279–320, Jun. 2022.

[43] M. M. Lau and K. H. Lim. *Shelly Chen Dataset 1*. Accessed: Oct. 2, 2023. [Online]. Available: https://github.com/SullyChen/driving-datasets

[44] M. M. Lau and K. H. Lim, "Convolutional and fully connected layer in DFN," *J. Inf. Sci. Eng.*, vol. 36, no. 5, pp. 1069–1078, 2020.

[45] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[46] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[47] R. Ma and S. Zhang, "An improved color image defogging algorithm using dark channel model and enhancing saturation," *Optik*, vol. 180, pp. 997–1000, Feb. 2019.

[48] S. McGrath, X. Zhao, R. Steele, B. D. Thombs, and A. Benedetti, "Estimating the sample mean and standard deviation from commonly reported quantiles in meta-analysis," *Stat. Methods Med. Res.*, vol. 29, no. 9, pp. 2520–2537, 2020.

[49] K. Miller, S. Chng, and L. Cheah, "Understanding acceptance of shared autonomous vehicles among people with different mobility and communication needs," *Travel Behaviour Soc.*, vol. 29, pp. 200–210, Oct. 2022.

[50] A. Mohammadi, K. Jamshidi, H. Shahbazi, and M. Rezaei, "Efficient deep steering control method for self-driving cars through feature density metric," *Neurocomputing*, vol. 515, pp. 107–120, Jan. 2023.

[51] N. Mostafa, H. S. M. Ramadan, and O. Elfarouk, "Renewable energy management in smart grids by using big data analytics and machine learning," *Mach. Learn. Appl.*, vol. 9, Sep. 2022, Art. no. 100363.

[52] F. Munir, S. Azam, B.-G. Lee, and M. Jeon, "Multi-modal fusion for sensorimotor coordination in steering angle prediction," 2022, *arXiv:2202.05500*.

[53] D. V. P. Mygapula, S. Adarsh, and K. P. Soman, "CNN based end to end learning steering angle prediction for autonomous electric vehicle," in *Proc. 4th Int. Conf. Electr., Comput. Commun. Technol. (ICECCT)*, Sep. 2021, pp. 1–6.

[54] Naoki, *Introduction to Udacity Self-Driving Car Simulator*, Feb. 2017. [Online]. Available: https://naokishibuya.medium.com/introduction-to-udacity-self-driving-car-simulator-4d78198d301d

[55] K. Nassiri and M. Akhloufi, "Transformer models used for text-based question answering systems," *Int. J. Speech Technol.*, vol. 53, no. 9, pp. 10602–10635, May 2023.

[56] E. Oğuz, A. Küçükmanisa, R. Duvar, and O. Urhan, "A deep learning based fast lane detection approach," *Chaos, Solitons Fractals*, vol. 155, Feb. 2022, Art. no. 111722.

[57] C. Oinar and E. Kim, "Self-driving car steering angle prediction: Let transformer be a car again," 2022, *arXiv:2204.12748*.

[58] M. O'Kelly, H. Abbas, and R. Mangharam, "Computer-aided design for safe autonomous vehicles," in *Proc. Resilience Week (RWS)*, Sep. 2017, pp. 90–96.

[59] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. Eur. Conf. Comput. Vis.*, Cham, Switzerland: Springer, 2006, pp. 430–443.

[60] M. Ryan, "The future of transportation: Ethical, legal, social and economic impacts of self-driving vehicles in the year 2025," *Sci. Eng. Ethics*, vol. 26, no. 3, pp. 1185–1208, Jun. 2020.

[61] H. Saleem, F. Riaz, L. Mostarda, M. A. Niazi, A. Rafiq, and S. Saeed, "Steering angle prediction techniques for autonomous ground vehicles: A review," *IEEE Access*, vol. 9, pp. 78567–78585, 2021.

[62] H. Saleem, F. Riaz, A. Shaikh, K. Rajab, A. Rajab, M. Akram, and M. S. Al Reshan, "Optimizing steering angle predictive convolutional neural network for autonomous car," *Comput., Mater. Continua*, vol. 71, no. 2, pp. 2285–2302, 2022.

[63] K. V. Savant, G. Meghana, G. Potnuru, and V. Bhavana, "Lane detection for autonomous cars using neural networks," in *Machine Learning and Autonomous Systems*. Singapore: Springer, 2022, pp. 193–207.

[64] M. Shahhosseini, G. Hu, and H. Pham, "Optimizing ensemble weights and hyperparameters of machine learning models for regression problems," *Mach. Learn. Appl.*, vol. 7, Mar. 2022, Art. no. 100251.

[65] R. Sharma and A. Sungheetha, "An efficient dimension reduction based fusion of CNN and SVM model for detection of abnormal incident in video surveillance," *June*, vol. 3, no. 2, pp. 55–69, May 2021.

[66] M. S. Shashank, N. T. Sri, S. Prathapaneni, M. Owais, M. Anil, and B. Saikumar, "Behavior cloning for self driving cars using attention models," vol. 6, no. 12, 2021.

[67] R. K. Tailor and S. Khan, "Robotic process automation (RPA) in the aviation sector," in *Global Air Transport Management and Reshaping Business Models for the New Era*. Hershey, PA, USA: IGI Global, 2022, pp. 289–300.

[68] M. Tezgider, B. Yildiz, and G. Aydin, "Text classification using improved bidirectional transformer," *Concurrency Comput., Pract. Exp.*, vol. 34, no. 9, p. e6486, Apr. 2022.

[69] T. Tian, C. Song, J. Ting, and H. Huang, "A French-to-English machine translation model using transformer network," *Proc. Comput. Sci.*, vol. 199, pp. 1438–1443, Jan. 2022.

[70] R. Valiente, M. Zaman, S. Ozer, and Y. P. Fallah, "Controlling steering angle for cooperative self-driving vehicles utilizing CNN and LSTM-based deep networks," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 2423–2428.

[71] J. Wang, Y. Chen, Z. Dong, and M. Gao, "Improved YOLOv5 network for real-time multi-scale traffic sign detection," in *Proc. Neural Comput. Appl.*, Dec. 2022, pp. 1–13.

[72] Q. Wang, W. Chen, X. Wu, and Z. Li, "Detail-enhanced multi-scale exposure fusion in YUV color space," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 8, pp. 2418–2429, Aug. 2020.

[73] Y. Wiseman, "Autonomous vehicles," in *Research Anthology on Cross-Disciplinary Designs and Applications of Automation*. Hershey, PA, USA: IGI Global, 2022, pp. 878–889.

[74] T. Wu, A. Luo, R. Huang, H. Cheng, and Y. Zhao, "End-to-end driving model for steering control of autonomous vehicles with future spatiotemporal features," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 950–955.

[75] Z. Xie, I. Sato, and M. Sugiyama, "Stable weight decay regularization," in *Proc. ICLR*, 2020.

[76] B. Yang, X. Song, Z. Gao, and N. Zhu, "Trajectory planning for vehicle collision avoidance imitating driver behavior," *Proc. Inst. Mech. Eng. D, J. Automobile Eng.*, vol. 236, no. 5, pp. 907–926, Apr. 2022.

[77] I. Yaqoob, L. U. Khan, S. A. Kazmi, M. Imran, N. Guizani, and C. S. Hong, "Autonomous driving cars in smart cities: Recent advances, requirements, and challenges," *IEEE Netw.*, vol. 34, no. 1, pp. 174–181, 2019.

[78] C. Ye, Y. Wang, Y. Wang, and M. Tie, "Steering angle prediction YOLOv5-based end-to-end adaptive neural network control for autonomous vehicles," *Proc. Inst. Mech. Eng. D, J. Automobile Eng.*, vol. 236, no. 9, pp. 1991–2011, Aug. 2022.

[79] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019.

[80] Y. Zeng, Z. Lin, J. Yang, J. Zhang, E. Shechtman, and H. Lu, "High-resolution image inpainting with iterative confidence feedback and guided upsampling," in *Proc. 16th Eur. Conf. Comput. Vis. (ECCV)*, Glasgow, U.K. Cham, Switzerland: Springer, Aug. 2020, pp. 1–17.

SUMMRA SALEEM received the bachelor's and master's degrees in computer science from the University of Engineering and Technology, Lahore, Pakistan. She is currently pursuing the Ph.D. degree in artificial intelligence with Rheinland-Pfälzische Technische Universität, Kaiserslautern, Germany. She is also with the Research Laboratory Learning from Test Data (FLaP)—a joint laboratory between IAV GmbH and German Research Center for Artificial Intelligence (DFKI GmbH), where she is most interested in deep learning for software engineering. Her research interests include natural language processing, pattern recognition, and probabilistic machine learning.

MUHAMMAD NABEEL ASIM received the Ph.D. degree in artificial intelligence from the Technical University of Kaiserslautern, Germany. Currently, he is a Senior Researcher with the Smart Data & Knowledge Services Unit, German Research Center for Artificial Intelligence (DFKI). He has published several papers in high impact factor journals and high ranked conferences. His research interests include natural language processing and bioinformatics.

LUDGER VAN ELST received the Diploma degree in computer science, economics, and mathematics from the University of Kaiserslautern. He is currently a Researcher, a Consultant, and the Project Manager for more than 20 years with the German Research Center for Artificial Intelligence (DFKI GmbH), Smart Data and Knowledge Services Group of Prof. Andreas Dengel. His scientific background is in the fields of semantic technologies, especially formal knowledge representation, ontologies, knowledge acquisition, and knowledge management. He has authored, coauthored, and edited more than 60 scientific publications on these topics. His current research interests and projects include semantic sensor data representation and analysis in industries 4.0 applications and in deep learning for applications in the automotive industry. His guiding vision is to develop technologies that better integrate data-driven and model-based approaches to create assistance systems that continuously learn and improve.

PETER SCHICHTEL received the Diploma degree in physics and the Dr. rer. nat. degree from the University of Heidelberg under the supervision of Dr. T. Plehn. His thesis topic was more jets in more LHC searches. He is currently the Team Lead of the Research Laboratory Learning from Test Data (FLaP)—a joint laboratory between IAV GmbH and German Research Center for Artificial Intelligence (DFKI GmbH), where he is most interested in time series analysis and deep neural network architecture. He was a member of the International Max Planck Research School for Precision Tests of Fundamental Symmetries, Heidelberg, Germany, and the Theoretical Advanced Study Institute of Elementary Particle Physics, University of Boulder, Boulder, CO, USA. He was appointed as an European Marie Curie Early Stage Researcher with the University of Durham, U.K., a member of the Monte-Carlo-Network, and a Visiting Scientist with CERN and Fermilab, IL, USA. He has over 20 publications in the fields of particle physics, statistical analysis, data mining, and time series analysis.

ANDREAS DENGEL received the Diploma degree in computer science from the University of Kaiserslautern and the Ph.D. degree from the University of Stuttgart. He is currently the Scientific Director with the German Research Center for Artificial Intelligence (DFKI GmbH), Kaiserslautern. In 1993, he became a Professor with the Computer Science Department, University of Kaiserslautern, where he holds the chair knowledge-based systems. Since 2009, he has been a Professor (Kyakuin) with the Department of Computer Science and Information Systems, Osaka Prefecture University. He was also with IBM, Siemens, and Xerox Parc. Moreover, he has coedited international computer science journals and has written or edited 12 books. He has authored more than 300 peer-reviewed scientific publications and has supervised more than 170 master's and Ph.D. theses. He is a member of several international advisory boards, has chaired major international conferences, and founded several successful start-up companies. He is a fellow of IAPR and has received prominent international awards. His main scientific emphasis is in the areas of pattern recognition, document understanding, information retrieval, multimedia mining, semantic technologies, and social media.