ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Multi-Task in Autonomous Driving through RDNet18-CA with LiSHTL-S Loss Function

Shang Shi[1] and Jian Qu[2]

ABSTRACT

Most current autonomous driving research focuses on single-task or dual-task methods. We propose to combine road tracking, obstacle avoidance, traffic sign recognition, and traffic light recognition in a single multi-task framework. Additionally, we validate it using a scale model car to confirm its viability in a semi-physical environment. We propose a novel framework, RDNet18-CA, designed to reduce the training requirements associated with enormous full-scene datasets. These massive full-scene datasets are utilized in the autonomous driving systems of companies such as Google and Tesla. Thus, our framework performs well with small training datasets and can function in unseen scenarios to a certain degree. Additionally, we present an innovative loss function, LiSHTL-S, that exhibits adaptivity. This allows the LiSHTL-S loss function to be dynamically modified based on the properties of the train data and the state of the model throughout the training phase, eliminating the requirement for intense manual parameter tuning. Lastly, we present a new traffic light design concept called the "traffic board" to enhance its resistance to lighting noise, making it more adaptable for autonomous driving. With these innovations in mind, our method outperforms existing methods in multiple areas.

Article information:

Keywords: Autonomous Driving System, RDNet18-CA, Road-Tracking, Obstacle Avoidance, Traffic Signs Recognition, Traffic Board, LiSHTL-S Loss Function

Article history:
Received: November 16, 2023
Revised: February 1, 2024
Accepted: March 28, 2024
Published: April 6, 2024 (Online)

DOI: 10.37936/ecti-cit.2024182.254780

1. INTRODUCTION

The research topic has grown explosively due to the rapid development of autonomous driving technology. Now, most research focuses on specific tasks, such as traffic sign recognition [1-3], traffic light recognition [4, 5], obstacle avoidance [6-8], and road tracking [9, 10]. Nevertheless, there is currently less focus on combining these tasks efficiently to create an entirely autonomous driving system. While Z. Nie and J. Qu [11] introduced a multi-task method that integrates both the MTS and MOD framework, it did not encompass the recognition of traffic light. Additionally, the range of obstacle avoidance tasks addressed in the method was relatively limited. This research presents a framework named RDNet18-CA (ResNet18 + Dropout Layer + Coordinate Attention Mechanism), which seeks to solve multi-task problems in autonomous driving systems, such as traffic sign recognition, traffic light recognition, obstacle avoidance, and road tracking.

Most commercial autonomous driving systems rely on obtaining, compiling, or accumulating environmental data to develop and train their systems. Environmental data is essential to autonomous driving systems, often requiring the training of the algorithm with massive amounts of data [12], such as street-view images and maps of an entire nation. This suggests that autonomous driving systems require a significant amount of environmental data for training to ensure their reliability across diverse scenarios. However, for humans to obtain a license of the driver, we don't need to practice driving on all the roads of a country. Therefore, mass data learning does not replicate how a human learns. Consequently, this research presents a novel method utilizing the LiSHTL-S (Linearly Scaled Hyperbolic Tangent Loss Scale) loss function combined with the RDNet18-CA framework. The aim is to reduce the size of the training dataset and achieve autonomous driving in unseen scenarios.

In the field of autonomous driving systems, researchers use four categories of environmental meth-

[1,2] The authors are the Faculty of Engineering and Technology, Panyapiwat Institute of Management, Nonthaburi, Thailand, E-mail: 6572100022@stu.pim.ac.th and jianqu@pim.ac.th
[2] Corresponding author: jianqu@pim.ac.th

ods, fully virtual, semi-simulation, semi-physical, and full-physical, to evaluate the performance of autonomous driving systems [13]. The fully virtual method performs computer simulations through virtual reality simulators or simulation software, which offer high safety and relatively low cost but are difficult to fully simulate the complexity of the real world due to the automatically simulated rewards. The semi-simulation method utilizes recordings from the real world to simulate real scenarios in virtual environments, obtaining a large amount of sample data from the recording. But it may not cover all situations, such as different lighting during the day. Both fully virtual and semi-simulation methods ignore the challenge of hardware problems in autonomous driving, especially for the semi-simulation method; the autonomous driving car cannot make free decisions due to the pre-recorded track. However, fully virtual and semi-simulation methods are relatively inexpensive and safe, especially for fully virtual methods. Semi-physical methods use scaled-down real elements and are tested in a laboratory environment using scale model cars, which are closer to real-world physical environments. Such a setup can simulate real-world lighting and shadowing noise and have a hardware setup similar to a real-world vehicle. On the other hand, the full-physical method is tested in the real world using actual cars and is possibly closest to real road and traffic conditions. It is one of the best for testing autonomous driving, but such a setup is costly and potentially dangerous to researchers, people, or animals in testing environments.

Whereas the semi-physical method is closer to the real-world physical method. It provides data more closely aligned with actual driving situations by using scale model cars for testing in the real world. Semi-physical methods allow for actual hardware and software validation. Finally, the relatively low cost and higher safety compared to real cars make the semi-physical method suitable for conducting autonomous driving research. Therefore, in this research, the semi-physical method was chosen.

In the field of autonomous driving, tasks are categorized into three levels: single-task, dual-task, and multi-task [14]. A single task focuses on solving a specific problem, such as road tracking or obstacle avoidance. Dual-task deals with two related tasks simultaneously, such as road tracking and traffic light recognition. Multi-tasking integrates multiple tasks, such as road tracking, obstacle avoidance, traffic signs, and traffic light recognition. The single-task is simple and focused, the dual-task considers the interconnection between tasks, and the multi-task maximizes the simulation of real driving scenarios. In this research, multi-tasking is chosen as the research object, specifically including road tracking, obstacle avoidance, traffic signs recognition, and traffic light recognition.

Finally, it is crucial to remain aware that modern traffic light is primarily made to be recognized by humans. However, the ability of autonomous driving systems to identify traffic lights depends on their ability to classify the red and green light image data acquired by cameras. Light noise from the sunshine can affect the image data that the camera captures [15], which could result in fluctuations in the red, yellow, and green light colors, which could affect the model recognition results. Therefore, to better satisfy the recognition requirements of autonomous driving systems, this research introduces a new traffic light design concept called the “traffic board.” The design is expected to enhance the ability of the traffic light to resist light interference, thereby improving the accuracy of traffic light recognition.

The following snippets describe the primary contributions of this research:

Implementing the RDNet18-CA Framework: In this research, we provide the RDNet18-CA framework, a novel autonomous driving framework. An attentional mechanism for managing road tracking, obstacle avoidance, traffic sign recognition, and traffic light recognition tasks in autonomous driving is integrated into this framework. Compared to existing methods, the framework not only accomplishes multitasking in autonomous driving but also demonstrates and performs better performance on small datasets.

New Traffic Light Design Concept: This research introduces the “traffic board,” a novel proposal for traffic light design. The objective is to optimize its suitability for autonomous driving cars rather than human vision. The newly designed “traffic board” viability in autonomous driving is tested in semi-physical environments.

Design of LiSHTL-S Loss Function: This research introduces a novel LiSHTL-S loss function that distinguishes itself from others by not requiring intensive manual parameter setting. Instead, it can be dynamically modified based on the properties of training data and the current state of the model, enabling the model to achieve a lower loss function value during training.

2. RELATED WORK

2.1 Deep Learning Architectures for Autonomous Driving Tasks

Al-Ni ma Raid Rafi Omar *et al.* [16] present an efficient method based on deep reinforcement learning for road tracking in autonomous car applications. This research proposes a new neural network that collects input states from a car-oriented forward view and generates appropriate road-tracking actions. Y. Li and J. Qu [17] propose a CNN model (PBLM-CNN21) with a novel architecture to enable real-time acceleration and deceleration while performing road tracks. H. Jian [18] implements autonomous tracking and obstacle avoidance using photoelectric sensors and ultrasound for obstacle avoidance. C. Zhou *et al.* [19] contrast with common obstacle avoidance modes based on a single sensor or solo algorithm; this article puts forward an intelligent pattern based on a combination of CNN-based deep learning and LiDAR-based image processing methods. H.-Y. Lin *et al.* [20] propose utilizing color information to precisely recognize traffic sign regions in an image. They employ neural networks for the recognition of traffic signs. N. Kanagaraj *et al.* [2] utilizes convolutional neural networks and spatial transformation networks for lane line detection and lenient-5 network architecture and Adam optimizer for traffic signs recognition to implement critical functions of semi-autonomous driving cars. Z. Nie and J. Qu [11] propose MTS and MOD models for road tracking, obstacle avoidance, and traffic sign recognition. J. Han [21] shows that existing autonomous driving systems rely heavily on "perfect" visual perception models trained using large amounts of annotated data to ensure safety. S. Ding and J. Qu [22] proposed the ENetb0-CBAM model, which utilizes attention mechanisms to enhance the environmental perception capabilities of autonomous cars, thereby reducing the likelihood of accidents. P. Sun *et al.* [23] introduce a new large-scale, high-quality, diverse dataset. The new dataset consists of 1150 scenes, each lasting 20 seconds, composed of high-quality LiDAR and camera data captured in a variety of urban and suburban geographic environments. Z. Bao *et al.* [24] propose that the high accuracy and information content of maps in localization rapidly make them a key component of autonomous driving.

2.2 Task Integration for Autonomous Driving System

Y. Li and J. Qu [25] proposed an end-to-end deep learning framework for multi-task autonomous driving. The research covers multi-tasking, such as road tracking, traffic sign recognition, and obstacle avoidance. Z. Nie and J. Qu [11] proposed an innovative network structure, MT-ResNet26, and the research also covered multi-tasks such as road tracking, traffic sign recognition, and obstacle avoidance; however, it is worth noting that the research did not include the recognition of traffic light.

In the research above, it is evident that research investigations on autonomous driving tasks are typically separated into single tasks. It is essential to keep in mind that the majority of this research concentrates on single-task, like obstacle avoidance, traffic signs recognition, traffic light recognition, and road tracking; they seldom ever incorporate multi-task combinations and, most of the time, do not include the task of traffic light recognition. This shows that in multi-task research related to autonomous driving, the issue of traffic light recognition is quite challenging. Thus, it cannot be easily incorporated into the autonomous driving model. However, the effectiveness and safety of autonomous driving systems depend on the ability to recognize traffic lights with reasonable accuracy. To close this research gap and enhance the overall performance and potential applications of autonomous driving systems, the research focuses on improving the integration and functionality of these systems, Such as traffic signs and traffic light recognition, obstacle avoidance, and road tracking.

2.3 Traffic Light Recognition

J. Nine and R. Mathavan [26] use the Mobile-Net-SSD model and transfer learning for real-time detection and recognition of traffic lights and brake lights. X. Wang *et al.* [15] propose that there are still many difficulties in traffic light recognition, such as the appearance of the traffic light, illumination, and inclement weather. D. Wang *et al.* [27] propose that it is challenging to collect and recognize traffic lights in various rare scenarios. S. Bali *et al.* [28] show that most algorithms perform better with the more considerable traffic light, but performance degrades with the smaller traffic light. R. Niroumand *et al.* [29] propose to add a new signal phase of the white phase based on the traditional traffic light to optimize the traffic control effect.

We have developed an innovative traffic light to improve adaptability to autonomous driving technology. To evaluate the performance of this new traffic light system, we conducted comparative experiments in different scenarios.

2.4 Loss Function-based Autonomous Driving

Y. Li and J. Qu [25] proposed an innovative multi-task autonomous driving loss function named MFPE (Multi-task autonomous driving-based loss Function). This loss function adopts a quadratic error relationship between the predicted and true values. Compared with the quadratic error relationship of the traditional MSE loss function, the quadratic error relationship is more effective in penalizing significant errors, thus avoiding the case of abnormally large single sample values. However, the parameter design of this loss function still relies on experience. In this context, S. Ding and J. Qu [30] proposed a new type of loss function called RE (Robustness Error), which improves the robustness of the model by keeping the output of the model less aggressively in the face of abnormal inputs due to the stricter constraints on the outliers. However, the RE loss function requires many data for tuning. To solve the above problems, this research proposes an innovative loss function LiSHTL-S, which aims to eliminate the requirement for intense manual parameter tuning. The design of this loss function takes into account the advantages of MFPE and RE without relying on too much experience or data for tuning and obtaining a lower loss value.

2.5 Privacy Protection and Solutions in Autonomous Driving Technologies

When applying autonomous car technology to the real world, privacy concerns may be related to collecting and processing sensitive data, such as location information, video footage of other vehicles and pedestrians, and sensor data. Several existing research efforts have proposed solutions for addressing these privacy problems.

S. Riyana and N. Riyana [31] introduced a privacy-preserving model to satisfy LKC privacy constraints through local and global data deletion, minimizing information loss while maintaining data utility. Experimental verification demonstrated its superiority over direct LKC applications, improving information loss and runtime performance without compromising privacy security. However, practical applications require consideration of additional factors like dynamics. S. Riyana [32] proposed a data dissection model addressing traditional anonymization model shortcomings by comparing multiple independently released versions of dynamic datasets. M.E. Nergiz *et al.* [33] suggested applying k-anonymity to trajectory data, demonstrating effectiveness in experiments with real and synthetic datasets. However, the utility metric only considers two dimensions, overlooking the complexity of application scenarios. These researches offer valuable insights into addressing privacy issues in autonomous driving systems while highlighting aspects that need further consideration in real-world applications.

3. MATERIALS AND METHODS

In this section, we initially describe the novel model framework and our proposed LiSHTL-S loss function, which enables us to perform the multi-task of autonomous driving. Then, new traffic lights appropriate for the artificial intelligence field are proposed and validated on a scale model car for its practicality in semi-physical environments.

3.1 RDNet18-CA Model Framework

This research proposes a novel RDNet18-CA framework to improve autonomous driving performance in unseen scenarios. The attention mechanism, which has been amply shown to enhance model performance, served as the basis for this research. In addition to inheriting the advantages of the ResNet18 model in image processing, it also adds the attention mechanism and dropout method to improve the performance of the autonomous driving system in unseen scenarios.

The architecture of the RDNet18-CA framework is depicted in Fig. 1. First, ResNet18 is an 18-layer neural network [34]. The ResNet18 architecture consists of one of 7×7 convolutional layer, a max-pooling layer, and four residual blocks, with each residual block comprising two or three of 3×3 convolutional layers. Finally, an average pooling layer and a classification layer are connected. A block-wise residual structure is used, which simultaneously retains the mobility of direct feature connectivity and reduces the number of parameters. ResNet18 employs a simplified residual block structure, where each residual block consists of one of a 3×3 convolutional layer, which is used to learn the spatial representation of the input features. After each convolutional layer, a BATCH normalization layer is introduced to speed up training and improve accuracy. A RELU function is added after each convolutional layer to enhance the nonlinear representation. Corresponding to the previous convolutional layer, the convolutional kernel size is 3×3. The input features are directly added to the output after the above operations to form the output of the residual block, and the final output of the residual block is equal to the sum of the mapping of the input features to the features learned in this propagation process.

Subsequently, an attention mechanism is added to the final layer in ResNet18 to enable precise localization and detection of marker positions. The framework may dynamically change the position of attention in response to various activities and input data because of the adaptive location-aware coordinate attention mechanism [35]. The coordinate attention module employs a strategic feature extraction method. The method utilizes a 1D pooling layer with a window size of (H,1) to pool the feature maps along the horizontal direction to extract the position information for each row of each channel and a 1D pooling layer with a window size of (1, W) to pool the feature maps along the vertical direction to extract the position information for each column of each channel. By decomposing the 2D pooling into two 1D pooling methods, the extraction of long-range dependencies in one direction and the retention of accurate positional information in the other direction are achieved.

The two sets of separately extracted features with position information are fused into a single feature map by a shared 1×1 convolution. Subsequently, the fused feature maps are separated into horizontal and vertical attention by independent 1×1 convolution, mapped to the attention weights using a Sigmoid function. Finally, the resulting horizontal and vertical attention are simultaneously applied to the input feature map to emphasize the representation of the region of interest. With this attention mechanism that combines positional information, the model can localize objects more accurately, which helps to improve the accuracy of object detection.

Finally, in the ResNet18 network architecture, the last layer is a linear classification layer, which directly

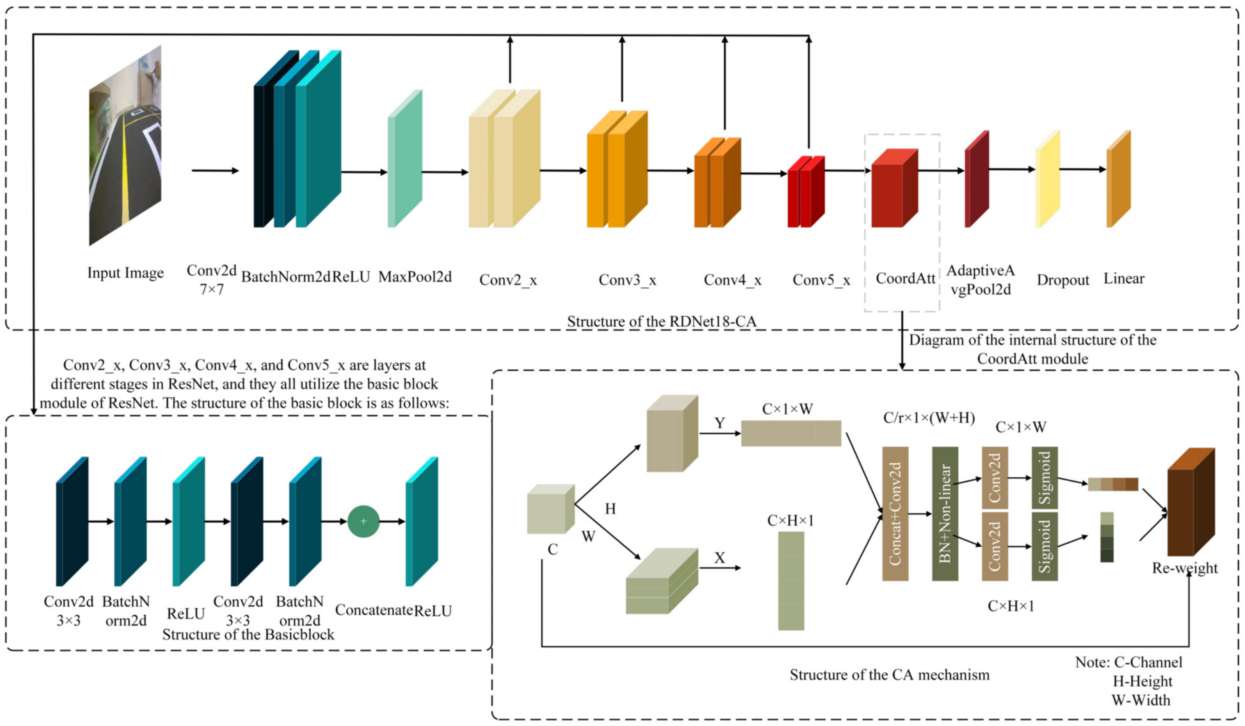
Input Image
Conv2d 7×7
BatchNorm2d
ReLU
MaxPool2d
Conv2_x
Conv3_x
Conv4_x
Conv5_x
CoordAtt
AdaptiveAvgPool2d
Dropout
Linear
Structure of the RDNet18-CA
Diagram of the internal structure of the CoordAtt module
Conv2_x, Conv3_x, Conv4_x, and Conv5_x are layers at different stages in ResNet, and they all utilize the basic block module of ResNet. The structure of the basic block is as follows:
Conv2d 3×3
BatchNorm2d
ReLU
Conv2d 3×3
BatchNorm2d
Concatenate
ReLU
Structure of the Basicblock
C×1×W
C/r×1×(W+H)
C×1×W
Y
H
W
C
X
C×H×1
Concat+Conv2d
BN+Non-linear
Conv2d
Conv2d
Sigmoid
Sigmoid
C×H×1
Re-weight
Structure of the CA mechanism
Note: C-Channel H-Height W-Width

Fig.1: *The structure of RDNet18-CA.*

maps features to the classification space for prediction. However, this layer tends to cause overfitting to the training data. Therefore, we introduce a Dropout layer before the last layer, which randomly masks the feature mapping, and a certain percentage of the outputs of neurons are set to 0. This operation helps to force the network to learn a broader range of feature representations, which reduces the risk of overfitting the training data.

3.2 LiSHTL-S Loss Function

The road-tracking task is regarded as a regression problem in this research. The Mean Square Error (MSE), Mean Absolute Error (MAE), and Huber regression loss functions, Robustness Error (RE) [30] are existing methods. Nevertheless, this research suggests a novel loss function, LiSHTL-S, to be dynamically modified based on the properties of the training data and the current state of the model, enabling the model to achieve a lower loss function value during training.

This research introduces a novel loss function, LiSHTL-S, by linearly scaling the tanh function to add more nonlinear components. The corresponding mathematical equation for LiSHTL-S is shown in Eq. (1). Its corresponding first-order derivative is represented by Eq. (2), and its corresponding second-order derivative is represented by Eq. (3). In addition to improving the ability of the model to adapt to different scenarios.

$$LiSHTL-S(x)=\frac{x}{scale}\times\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{1}$$

$$LiSHTL-S'(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}+\frac{4x}{(e^{x}+e^{-x})^{2}} \tag{2}$$

$$LiSHTL-S''(x)=\frac{8}{(e^{x}+e^{-x})^{2}}+\frac{8x\times(e^{-2x}-e^{2x})}{(e^{x}+e^{-x})^{4}} \tag{3}$$

The LiSHTL-S loss function includes a learnable parameter named “scale” in particular. Throughout the training phase, this value is gradually tuned to provide the LiSHTL-S loss function with an adaptable quality. When learning first begins, “scale” may have a high value that makes it insensitive to outliers and allows the model to fit normal samples more accurately. The “scale” is gradually decreased during training to match the distribution of normal samples better.

When the model matches the data distribution, the “scale” may be raised again to reduce the effect of outliers on the performance of the model. During the training phase, this adaptability enables the LiSHTL-S loss function to be dynamically altered by the properties of the data and the state of the model without manually configuring the parameters.

We consider scale as a hyper-parameter to be learned, which is initialized as “1”. Specifically, the scalar value is defined as a torch parameter and added to the optimization variables of the optimizer. The

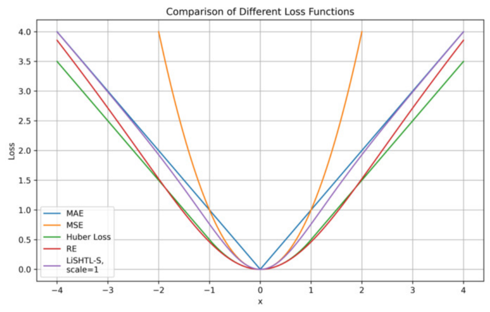

Fig.2: *Comparison of Different Loss Functions.*

adaptive adjustment of hyper-parameters enhances the efficiency of the model during training, enabling faster convergence and better performance. Simultaneously, this design reduces the need for manual adjustments, alleviating the burdensome task of manual tuning.

The performance of several other loss functions, including MAE, MSE, Huber, and RE [30], is compared to demonstrate the usefulness of the presented LiSHTL-S loss function. Fig. 2 plots the original images of MSE, MAE, Huber, RE, and LiSHTL-S images. The images of the original loss functions demonstrate that LiSHTL-S reaches the MAE more quickly and exhibits greater resilience than RE.

The loss function reflects the difference between the true and predicted values. When the error is significant, the MSE causes the error to expand quadratically, which sets off the gradient explosion issue. This makes these locations with substantial mistakes, which are sometimes referred to as outliers, and may interfere with the model; such outliers will be more noticeable to the model. LiSHTL-S utilizes negative value information more effectively than MSE and is less vulnerable to the gradient vanishing issue. Additionally, LiSHTL-S can prevent the gradient from vanishing in the negative value area, which has a significant error compared to MAE. Compared to Huber, the gradient of the model is more evenly distributed over the negative region, reducing the likelihood of encountering extreme gradient values.

In comparison to the frequently used squared loss, the LiSHTL-S loss function has some noteworthy characteristics, such as being unbounded and symmetric, which better utilize the negative value information. Additionally, Fig. 3 represents the first-order derivatives corresponding to the loss function. Fig. 4 represents the second-order derivatives corresponding to the loss function. The derivative of the LiSHTL-S does not reach zero, which aids in preventing the gradient vanishing issue and makes it easier to train models. Finally, the loss descent process is theoretically accelerated by the second-order derivative of LiSHTL-S being extremely sensitive to changes in the input values.

3.3 Traffic Board

Traffic light is primarily made to make it easier for people to evaluate traffic conditions and choose the best course of action for controlling cars. Fig. 5 shows traffic light. The conventional traffic light design, however, may introduce physical interference for autonomous driving systems that use cameras to collect image data and make traffic light judgments through deep learning models, leading to errors in the judgment of the model as autonomous driving technology gains in popularity.

When taking photos of conventional traffic lights in the scenarios, the camera is vulnerable to several physical interferences that could lead to mistakes in the decision-making of the model. This research suggests a novel method for designing traffic lights utilizing a traffic board instead of conventional colored lights, known as a "traffic board." This design mitigates physical interferences and remains effective in low-light conditions (such as nighttime or roads without streetlights). Figure 6 illustrates the images of these traffic boards.

4. EXPERIMENTAL SETUP

In this section, we elaborate on how to build an autonomous driving platform and explain the application of deep learning in realizing multi-tasks for autonomous driving. Then, we provide a new benchmark for measuring the effectiveness of models and explain how to determine which models perform better.

4.1 Autonomous Driving Platforms

Jetbot, an intelligent vehicle built on the free and open-source Jetbot AI robotics platform, served as the experimental car for this research. The scale model car is shown in Fig. 7. The all-wheel-drive system Jetbot uses comprises DC-geared motors. Basic vehicle motions such as straight ahead, left, right, and stop can be achieved by varying the speed value of each motor. In this research, the output commands of the model are utilized to control the car.

Table 1: *Data collection.*

	Raspberry Pi 4B	Jetson nano
CPU	ARM Cortex-A72 Quad-core 1.5GHz	Quad-core ARM A57 @ 1.43 GHz
CPU	Broadcom VideoCore IV@500MHz	128-core Maxwell
RAM	Choice of 2GB/4GB/8GB	4 GB 64-bit LPDDR4 25.6 GB/s
USB	2x USB2.0 + 2x USB3.0	4x USB 3.0, USB 2.0 Micro-B
Camera	MIPI CSI port*1	MIPI CSI port*2
Other	40-pin GPIO	40-pin GPIO

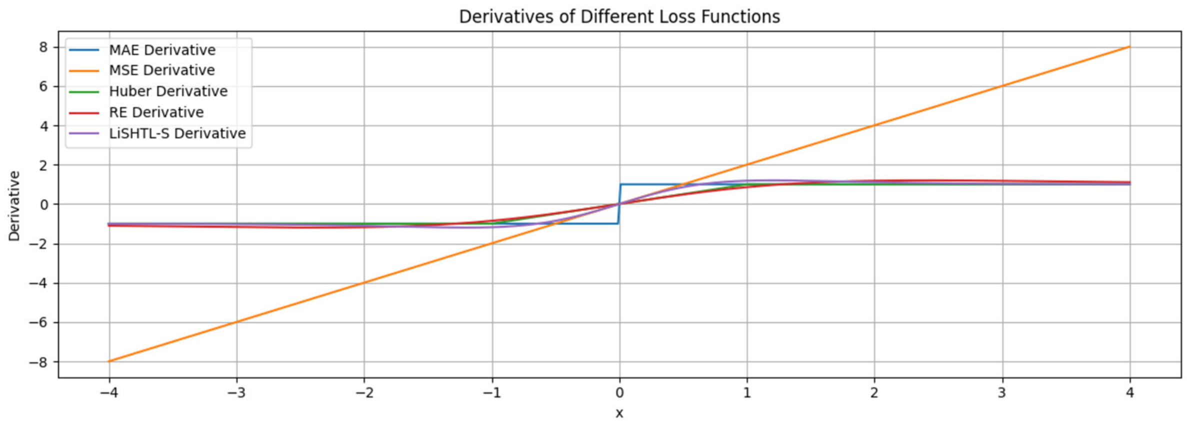

Fig.3: *Derivatives of Different Loss Functions.*

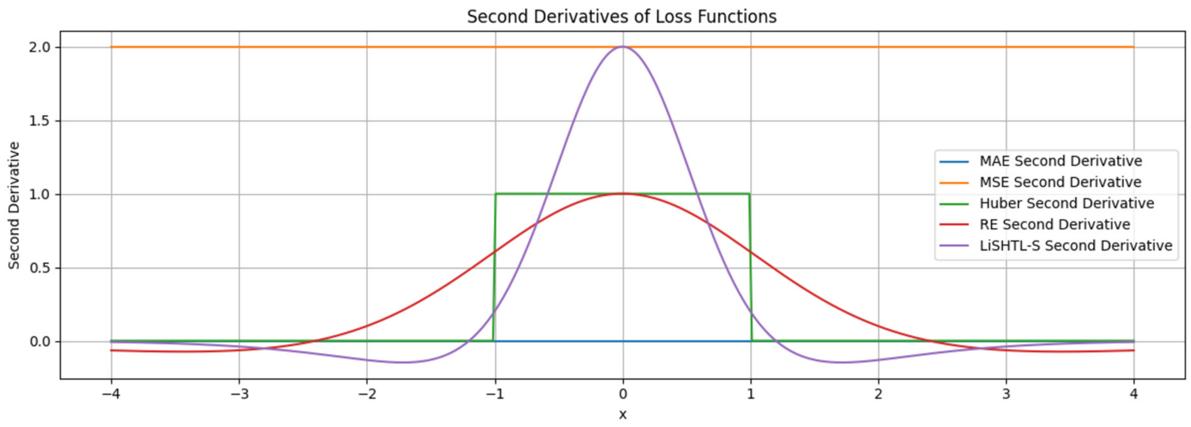

Fig.4: *Second Derivatives of Loss Functions.*

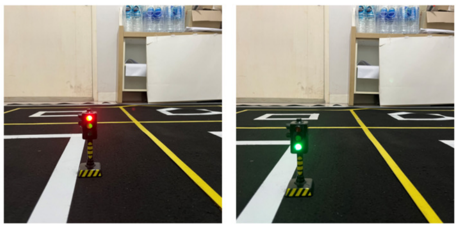

Fig.5: *The red light and the green light are activated.*

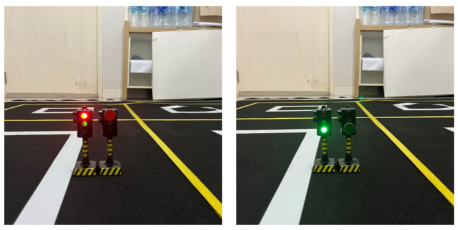

Fig.6: *The red board and the green board are activated.*

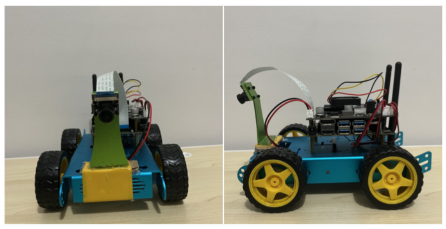

Fig.7: *Front View and Left View of the Jetbot.*

An IMX219-160 camera was employed in this research as the image input source for data collection and model testing. The platform for model computation was decided to be the Jetson Nano. Even though the Jetson Nano and Raspberry Pi 4B have many features in common, the Jetson Nano has a more potent GPU. The GPU of Raspberry Pi 4B is primarily used to support display output and simple graphics applications. Jetson Nano has a GPU that excels at deep learning and computer vision tasks, with much higher performance than the Raspberry Pi 4B. Jetson nano supports CUDA and cuDNN for inference of deep learning models. A comparison of Jetson Nano and Raspberry parameters is shown in Table 1.

4.2 Track Design

Since the experimental data collection must be done on the track, the track design is essential to the research. Most tracks offered by the existing open-source project platforms are circular and only

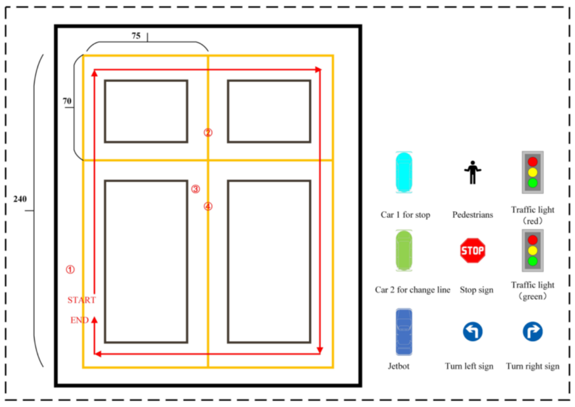

Fig. 8: Track 0

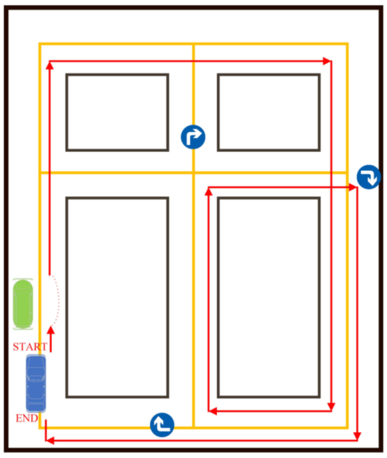

Fig. 9: Track 1

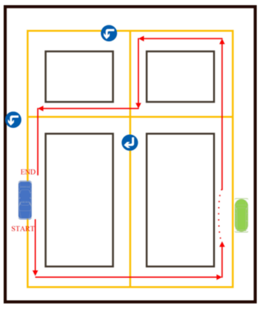

Fig. 10: Track 2

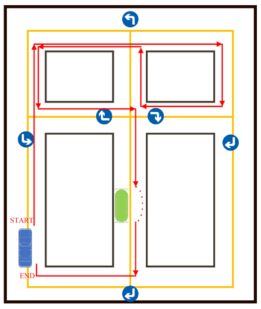

Fig. 11: Track 3

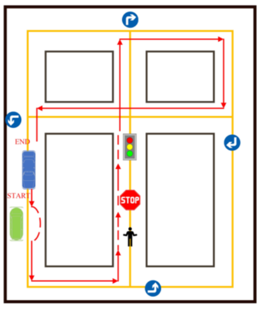

Fig. 12: Track 4

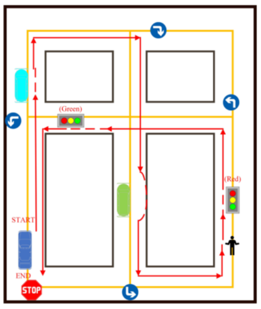

Fig. 13: Track 5

helpful in completing road tracking tasks; this research chooses to reconstruct the track to replicate many of the full-physical environments. The new track has straights, turns, T-intersections, and cross-intersections to accommodate the needs of the experiments and more accurately represent real-world driving scenarios. The precise design is shown below.

Track 0: Track 0 is shown in Fig. 8. Track 0 uses the outside circle as a predetermined route for gathering road tracking images. At fixation 1, we collect images of the vehicle used to avoid obstacles; at fixation 2, we record images of the left and right turn signals; at fixation 3, we gather images of the traffic light; and finally, at fixation 4, we gather images of the STOP sign and pedestrians. Track 0 is for training data acquisition.

Track 1: Track 1 is shown in Fig. 9. Track 1 is used for testing against a model trained from Track 0.

Track 2: Track 2 is shown in Fig. 10. Track 2 is used for testing against a model trained from Track 0. Track 2 involves a counterclockwise direction, which checks if the clockwise direction recording done in Track 0 can be applied in Track 2.

Track 3: Track 3 is shown in Fig. 11. Track 3 is used for testing against a model trained from Track 0. Track 3 introduces an inner circle route in both the clockwise and counterclockwise directions. We randomly place traffic signs over the track for the experiment.

Track 4: Track 4 is shown in Fig. 12. Track 4 is used for testing against a model trained from Track 0. We introduce traffic signs, traffic lights, and obstacle avoidance into the experiment.

Track 5: Track 5 is shown in Fig. 13. Track 5 is used for testing against a model trained from Track 0. To evaluate the fusion of the model, recognition capability for unseen scenarios, and anti-interference abilities, track 5 randomly places traffic signs, traffic lights, and obstacle avoidance signs at various points on the track.

4.3 Data Collection

The task was decomposed into five primary subtasks for data collection: lane keeping, left and right sign turning, lane changing, road unobstructed (no obstructions, green light), and road obstructed (pedestrians, red light, STOP sign). This breakdown enables a more thorough model performance analysis in various driving scenarios.

In this research, we constructed the dataset by capturing image data through the camera of a scale model car and labeling the images as they are captured; for example, the image data for road tracking, when captured through the camera, is labeled with the x and y coordinate values. The images were all 224×224 RGB images with a constant resolution throughout the data collection process. Because of this consistency, the model can process input data of similar resolution for various positions.

Each image has a label that follows a specific naming scheme. To accurately match the data during model training and testing, in this instance, X and Y stand in for the steering angle of the Jetbot. Each image is also given a distinct “uuid1” to ensure the data is unique. Details of the road tracking dataset are shown in Table 3.

Labels such as “Obstructed” and “Unobstructed” represent the context for the obstacle avoidance task. These labels make it easier for us to assess the performance of the model throughout the testing process and ensure it can react when facing various road conditions. Details of the obstacle avoidance dataset are shown in Table 4.

4.4 Experimental Parameter Setting

In this research, we use the Google Colab for training. The organized dataset is uploaded to Google Drive for model training. We adopted the PyTorch framework, specifically torch 1.11.0, torchvision 0.12.0, Python 3.11, and CUDA 12.1. By training in Colab. Finally, we deployed the trained models to the Jetson Nano for model testing and performance evaluation.

A more accurate assessment of the performance of the model can result from selecting an appropriate partitioning strategy for various tasks and data scenarios. A “hold-out” split is chosen for the road tracking task since it is regarded as a regression problem and there are enough datasets. Because obstacle avoidance, traffic signs, and traffic light recognition tasks involve classification and there is small data in each category, it might not be possible to assess the performance of the model with enough accuracy. The “k-folded” partition is selected in the present instance.

The road tracking dataset is divided into training and testing sets, with the ratio being 8:2, the epoch setting being 80, the learning rate being 0.001, and the optimizer chosen as Adam.

The dataset uses K-folded cross-validation for obstacle avoidance, traffic signs, and traffic light recognition tasks. The ideal number of folds can be chosen from a variety of possibilities. There is no requirement to consider the number of folds if the sample size is enormous (> 10k, > 100k). The data should be divided into 10 or 20 binary numbers. If the sample size is small, a straightforward rule like the rule of storage can be used to determine the correct number of folds. This research uses Sturge regulations [36] to calculate the appropriate number of folds. The corresponding equation is shown in Eq. (4). Where N is the number of samples in the dataset. The dataset is divided into 12 folds in this research as obtained by the equation. We set the epoch to 10, the learning rate to 0.001, the momentum to 0.9, and the optimizer to SGD.

$$Number\ of\ Folds \ = \ 1 + \log_2(N) \tag{4}$$

4.5 Evaluation Criteria

S. Ding and J. Qu [11] employ the accident rate as an evaluation criterion, where the rate represents the proportion of instances of rule violations to the total number of tests on simulated roads. They conducted multiple experiments on autonomous cars when facing obstacles. If the autonomous vehicle collides with or deviates from the road during testing, it is deemed a rule violation. The evaluation criteria of the Carla platform primarily emphasize the completion of driving tasks, overlooking fundamental road tracking assessments in autonomous driving tasks. Therefore, we propose a novel method for autonomous driving evaluation, focusing on route completion (RC) and infraction penalties (IP) for driving rule adherence. During the evaluation process, it is essential to validate the performance of the model on the driving task using a variety of measures. The driving score (DS) is the primary statistic used in the road tracking task; a higher driving score indicates a better model, and it can be determined using Eq. (5) below:

$$DS \ = \ \frac{1}{N_c}\sum_{i=1}^{N_c} RC_i(1 - IP_i) \tag{5}$$

N_c : Number of Test Lap
i : -th route
RC_i : the percentage of completion of the route
IP_i : the infraction penalty for a route

The final driving score, DS, is calculated by multiplying the tolerance of route “i” by its percentage of route completion, dividing it by the total number of tests, and taking the mean value. RC is measured as a percentage of its overall length—the infraction penalty of the route. Meanwhile, RC can be calculated using Eq. (6). IP can be calculated using Eq. (7).

$$RC_i = \frac{CD}{TD} * 100\% \tag{6}$$

CD : Completed Distance
TD : Total Distance
i : -th route

$$IP_i = \frac{1}{10}\sum_{j}^{M} p_j \tag{7}$$

j : denotes the penalty corresponding
p_j : represents the penalty score
M : represents the number of types of penalty behaviors considered in the evaluation process

IP sums together all of the route "i" infraction scores and divide them by the total base score. The grading process also considers the violation dataset V (Table 2), which contains various infractions. Each infraction has a penalty score, usually starting at 10, and depending on the severity of the infraction, a different penalty score is issued. Specific penalty scores are shown in Table 2.

Table 2: *Types of Penalties and Scores.*

Infractions	Infraction penalty
Crossing Lane	-1
Going Off-road	-2
Mistake in a Turn sign	-2
Failed in Obstacle Recognition	-2
Failed in Vehicle Avoidance	-3

To properly evaluate the performance of the model in the driving task, this method of generating the driving score enables the combination of route completion and infractions to be considered. The model performs better with a higher driving score.

The confusion matrix and associated metrics are used to assess the classification performance of the model in the obstacle avoidance task. The classification accuracy of the model is summarized using the confusion matrix, which contains true positive examples (TP), true negative examples (TN), false positive examples (FP), and false negative examples (FN). When the category distribution is imbalanced, metrics like accuracy, precision, recall, and F1 score (Eqs.8-11) are used to assess the performance of the model across many categories, and these metrics can offer a more accurate performance assessment.

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{8}$$

$$precision = \frac{TP}{TP + FP} \tag{9}$$

$$recall = \frac{TP}{TP + FN} \tag{10}$$

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{11}$$

5. RESULTS AND DISCUSSION

In this section, we explain the experimental results in detail. Especially about the analytical results for various tasks of autonomous driving based on specific evaluation criteria and compared them with existing methods.

5.1 Effects of Data Set Size and Evaluation in Unseen Scenarios

In this experiment, various dataset sizes are used to test the performance of the model on the three tracks that have been constructed. To evaluate the performance of the proposed RDNet18-CA model with small datasets, the research involved comparing models trained with different dataset sizes and applying them to a scale model car to perform autonomous road tracking tasks. DS is used to evaluate the strengths and weaknesses of each model. Table 5 shows that the DS score of each model improves as the dataset increases. However, compared to the ResNet18, its DS average is 0.27 for a dataset size of 1500 and 0.88 for a dataset size of 2000. this shows that different dataset sizes significantly impact the model performance of ResNe18. It is worth noting that by comparing the RDNet18-CA model proposed in this research, its DS mean value is 0.78 when the dataset size is 1500, which shows the improvement over the ResNet18 model in small datasets.

Also, the research used the MT-ResNet26 model as a reference model for comparison. According to the experimental data, When the dataset size is 1500, both MT-ResNet26 [11] and ResNet18 models produce average DS scores of 0.27, but our RDNet18-CA model produces an average DS score of 0.88. This shows that RDNet18-CA performs better in small datasets than other models.

In Table 5, the research also compares the performance of the scale model cars in different scenarios. Among them, the "track0" scene is the collection of

Table 3: *The Dataset of Road Tracking obtained from track 0.*

Dataset	Road Tracking	Turn Right/Left	Barrier Avoidance Vehicles	Total Number
Dataset1	500	500	500	1500
Dataset2	600	500	900	2000
Dataset3	2000	1400	1400	4800

Table 4: *The Dataset of Obstacle Avoidance, Traffic Signs, And Traffic Light Recognition obtained from track 0.*

Dataset	Traffic light(red)	Pedestrians	Stop sign	Vehicle	Traffic light(green)	Road tracking
Dataset4	600	200	100	100	600	300

Table 5: *Results of The Road Tracking Experiment.*

Dataset	Model	Track	DS	RC	IP
Dataset1 (1500)	ResNet-18	Track1	0.414	69%	40%
		Track2	0.163	19%	14%
		Track3	0.241	45%	48%
	MT-ResNet26 [11]	Track1	0.114	69%	40%
		Track2	0.568	71%	20%
		Track3	0.129	23%	44%
	RDNet18-CA	Track1	0.677	82%	18%
		Track2	0.885	96%	8%
		Track3	0.774	86%	14%
Dataset2 (2000)	ResNet-18	Track1	0.96	100%	4%
		Track2	0.9	100%	10%
		Track3	0.79	85%	18%
	MT-ResNet26 [11]	Track1	0.381	56%	32%
		Track2	0.64	80%	20%
		Track3	0.399	57%	30%
	RDNet18-CA	Track1	0.86	100%	14%
		Track2	0.96	100%	4%
		Track3	0.8	90%	12%
Dataset3 (4800)	ResNet-18	Track1	0.91	94%	4%
		Track2	0.34	90%	8%
		Track3	0.94	94%	6%
	MT-ResNet26 [11]	Track1	0.6	36%	20%
		Track2	0.88	100%	12%
		Track3	0.49	70%	30%
	RDNet18-CA	Track1	0.96	100%	4%
		Track2	0.65	68%	4%
		Track3	0.98	100%	2%
Notes: Max RC: 100%, Ideal RC: 100% Max IP: 100%, Ideal IP: 0%					

Table 6: *Results of The Obstacle Avoidance, Traffic Signs, and Traffic Light Recognition Experiment.*

Dataset	Model	Scenario	Accuracy	Recall	Precision	F1
Dataset4 (1900)	**RDNet18-CA**	Normal	100%	100%	100%	1
		Normal+light	98%	100%	95%	0.97
		Noramal+natural light	96%	100%	93%	0.96
Dataset4 (1900)	ResNet18	Normal	91%	100%	85%	0.92
		Normal+light	86%	100%	78%	0.88
		Noramal+natural light	88%	100%	80%	0.89
Dataset4 (1900)	MT-ResNet26 [11]	Normal	98%	100%	95%	0.97
		Normal+light	95%	100%	91%	0.95
		Noramal+natural light	88%	100%	80%	0.89
Ideal Result			100%	100%	100%	1

images used for training each model, while "track1," "track2," and "track3" incorporate unseen images to test the applicability of the model under unseen scenarios.

From Table 5, it can be observed that in "track1," "track2," and "track3," the ResNet18 and MT-ResNet26 models can accomplish the task in some of the unseen scenarios to a certain extent. However, in contrast, the RDNet18-CA framework proposed in this research performs much better in these unseen scenarios, with a mean completion of 89% and a penalty of 3% under the comparison of dataset 3. By looking at the values of RC and IP, it can be concluded that RDNet18-CA has some ability to cope with autonomous tasks in unseen scenarios.

5.2 Single-Task Evaluation

This experiment assesses the ability of the model to recognize obstacles in a single task and the impact of light interference. Recreating various obstacle scenarios involves different test conditions, encompass-

Table 7: *Results of The Road Tracking, Obstacle Avoidance, Traffic Signs, and Traffic Light Recognition Experiment.*

Dataset	Model	Track	DS	RC	IP
Dataset2	MobileNet	Track4	0.68	54%	20%
		Track5	0.72	80%	12%
Dataset2	ResNet18	Track4	0.88	92%	12%
		Track5	0.34	34%	0%
Dataset2	EfficientNet	Track4	0.94	100%	6%
		Track5	0.27	34%	20%
Dataset2	MT-ResNet26 [11]	Track4	0.74	100%	5%
		Track5	0.48	60%	20%
Dataset2	ENetb0-CBAM [20]	Track4	0.76	86%	12%
		Track5	0.64	76%	16%
Dataset2	**RDNet18-CA**	Track4	0.94	100%	6%
		Track5	0.92	100%	8%
Notes: Max RC: 100%, Ideal RC: 100% Max IP: 100%, Ideal IP: 0%					

Table 8: *Comparison of Loss Functions.*

Loss	Epoch	Loss Value
MSE	29	0.002259
MAE	74	0.025015
Huber	60	0.001268
RE [30]	79	0.000884
LiSHTL-S	60	0.000331

Table 9: *Results of Deployed Loss Function in Unseen Scenarios.*

Loss	DS	RC	IP
MSE	0.3	50%	40%
LiSHTL-S	0.7	100%	30%
Ideal Result	1	100%	0

Table 10: *Results of The Traffic Light vs Traffic Boards.*

Scenario	Type	Red	Green	Accuracy
Normal	Traffic light	2	10	60%
	Traffic board	10	10	100%
Normal+lighting	Traffic light	0	10	50%
	Traffic board	8	10	90%
Normal+natural light	Traffic light	2	10	60%
	Traffic board	9	10	95%
Ideal Result		10	10	100%

Table 11: *Comparison of Traffic Board in Daytime and Nighttime.*

Scenario	Type	Red	Green	Accuracy
Daytime	Traffic Board	10	10	100%
Nighttime		8	10	90%
Ideal Result		10	10	100%

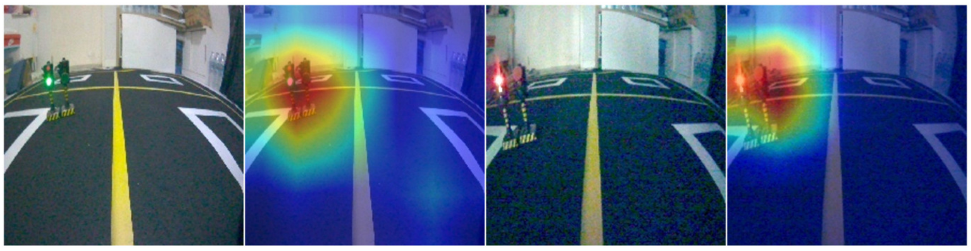

Fig.14: *Traffic Board-to-Heatmap.*

ing pedestrians, traffic signs, and traffic lights. The effectiveness of the model is subsequently evaluated, with obstacle recognition and overall performance in the obstacle avoidance task being potential evaluation criteria.

Table 6 indicates that each model can successfully recognize obstacles, but variations exist among them. For instance, after conducting ten experiments under the same setting, the statistical results reveal an average accuracy of 91% for ResNet18 and 98% for MT-ResNet26 [11], and our proposed model achieves perfect recognition.

In Table 6, we can observe that the light interference leads to decreased recognition accuracy. Meanwhile, the average recognition rate under light interference is 88% for ResNet18, 94% for MT-ResNet26, and 98% for RDNet18-CA.

5.3 Multi-Task Evaluation

In Table 7, a comparison of the scores of scale model cars on the track under different models is conducted. This includes some standard deep learning models such as MobileNet, ResNet18, etc., and some models published by related researchers, such as MT-ResNet26 [11] and ENetb0-CBAM [20]. Multi-tasking in autonomous driving is covered in Table 7, including the evaluation of RC for autonomous driving road tracking tasks and the evaluation of IP for traffic signs recognition, obstacle recognition, vehicle avoidance, and traffic light recognition tasks.

Table 7 illustrates how the trained model obtained from the road tracking experiment was used as the testing model in the road tracking and obstacle avoidance experiment and how it was combined with the classification model to control the autonomous car and complete the multi-task. Compared to the existing model, the MT-ResNet26 model [11] performed very well in Track 4, and MobileNet performed very well in Track 5. However, there was a significant difference in the degree of completion between the two models.

By observing Table 7, it can be seen that the models proposed by existing research, such as MT-ResNet26 and ENetb0-CBAM [20], have improved their mean values compared to standard models such as MobileNet, ResNet18, and EfficientNet. ResNet18 and MT-ResNet26 have an average value of 0.61, ENetb0-CBAM has an average value of 0.7, while RDNet18-CA achieves an average value of 0.93.

5.4 Loss Function Evaluation

By comparing the results in Table 8, we observed that MSE obtained the optimal model at epoch 29 under the same dataset and parameter settings with a loss value of 0.002259. In comparison to MAE, Huber, RE [30], and LiSHTL-S, both LiSHTL-S and Huber achieved the optimal model at epoch 60, yet the loss value of Huber was 0.001268, while the loss value of LiSHTL-S was 0.000331. Additionally, compared to the RE loss function, RE demonstrated improvements over MSE, MAE, and Huber. However, LiSHTL-S performs better in both the number of epochs and loss values. Therefore, our proposed LiSHTL-S, compared to existing loss functions, demonstrates the ability to attain the optimal model more swiftly and achieve lower loss values.

In Table 9, we deployed standardized MSE and LiSHTL-S loss functions on the scale model car and evaluated their performance in semi-physical. We observed a D.S. value of 0.3 for MSE and 0.7 for LiSHTL-S during the experiment. This indicates that LiSHTL-S allows the model to achieve lower loss values through scale adjustment via hyper-parameters. Therefore, the model car exhibits improved performance in semi-physical, showcasing its capability to handle unseen scenarios.

5.5 Experiment on the Impact of Lighting Conditions on Traffic Light

The data comparison in Table 10 reveals that the camera of the scale model car often makes errors in recognizing red lights during the real-time identification of conventional traffic lights. Furthermore, the proposed traffic sign recognition method in this research remains reliable in different light environments, demonstrating resilience against light interference.

The purpose of the design in Table 11 is to assess the ability of deep learning models to recognize traffic board in both nighttime and daytime environments. This indicates that our approach can adapt to different lighting conditions for traffic board under specific circumstances.

In Fig. 14, we can see the focus of the deep learning model on the traffic board. The performance of this attention not only highlights the accurate perception of the target by the model but also reveals the ability of the model to adapt to traffic scenarios in both nighttime and daytime environments.

6. CONCLUSIONS

This research effectively accomplishes multi-tasks such as road tracking, obstacle avoidance, traffic sign recognition, and traffic light recognition with a single autonomous driving framework. This research introduces a novel RDNet18-CA framework, which integrates attention and dropout mechanisms with our newly proposed LiSHTL-S loss function. This combination aims to achieve autonomous driving tasks with a small dataset. We conducted examinations to confirm the viability of our proposed method in semi-physical environments with a scale model car. Additionally, the framework demonstrates effective adaptability to handle some unseen scenarios. The results from Table 5 indicate that our proposed framework accomplishes road-tracking tasks with a small dataset and effectively adapts to unseen scenarios along the given track. Moreover, the results of Table 7 showed that the average driving score for RDNet18-CA was 0.93, while the average driving score for ResNet18 was 0.61, and the average driving score for ENetb0-CBAM [20] was 0.7. The results indicate that the algorithm can achieve notable gains in road tracking, obstacle avoidance, traffic signs, and traffic light recognition compared to other models. Additionally, this research introduces and compares a novel loss function with commonly used loss functions in regression problems. The results in Table 8 show that the LiSHTL-S loss function achieves a loss value of 0.000331, significantly lower than MSE and Huber and RE [30]. Moreover, LiSHTL-S demonstrates faster convergence in earlier epochs. Lastly, this research proposes a traffic light design for autonomous driving. The results of Table 10 for the traffic board show that the newly designed traffic signs (traffic board) achieve 95% recognition accuracy compared to 56% for the traffic light.

The source code and project video of this research can be found at `https://github.com/eamonn-ss/Jetbot_Integrated-Traffic-Intelligence-Task`.

ACKNOWLEDGEMENT

The first and last authors each contributed 50% equally to this work. The first author is awarded a full scholarship from CPALL for conducting this research in PIM. The second author is the corresponding author.

AUTHOR CONTRIBUTIONS

Conceptualization, S.S. and J. Q.; methodology, S.S. and J. Q.; software, S.S. and J. Q.; validation, S.S. and J. Q.; formal analysis, S.S. and J. Q.; investigation, S.S. and J. Q.; data curation, S.S. and J. Q.; writing—original draft preparation, S.S. and J. Q.; writing—review and editing, S.S. and J. Q.; visualization, S.S. and J. Q.; supervision, J. Q.; All authors have read and agreed to the published version of the manuscript.

References

[1] Y. Satılmış, F. Tufan, M. Şara, M. Karslı, S. Eken and A. Sayar, "CNN Based Traffic Sign Recognition for Mini Autonomous Vehicles," *Information Systems Architecture and Technology: Proceedings of 39th International Conference on Information Systems Architecture and Technology – ISAT*, pp. 85-94, 2019.

[2] N. Kanagaraj, D. Hicks, A. Goyal, S. Tiwari and G. Singh, "Deep learning using computer vision in self-driving cars for lane and traffic sign detection," *International Journal of System Assurance Engineering and Management*, vol. 12, no.6, pp. 1011-1025, 2021.

[3] S.-C. Huang, H.-Y. Lin and C.-C. Chang, "An In-Car Camera System for Traffic Sign Detection and Recognition," *2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS)*, pp. 1-6, 2017.

[4] J. Kim, H. Cho, M. Hwangbo and J. Choi, "Deep Traffic Light Detection for Self-driving Cars from a Large-scale Dataset," *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 280-285, 2018.

[5] L. C. Possatti, R. Guidolini, V. B. Cardoso, R. F. Berriel, T. M. Paixão, C. Badue, A. F. De Souza and T. Oliveira-Santos, "Traffic Light Recognition Using Deep Learning and Prior Maps for Autonomous Cars," *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1-8, 2019.

[6] X. Zong, G. Xu, G. Yu, H. Su and C. Hu, "Obstacle Avoidance for Self-Driving Vehicle with Reinforcement Learning," *SAE International Journal of Passenger Cars - Electronic and Electrical Systems*, vol. 11, no.1, pp. 30-39, 2017.

[7] Z. F. Li, J. T. Li, X. F. Li, Y. J. Yang, J. Xiao and B. W. Xu, "Intelligent Tracking Obstacle Avoidance Wheel Robot Based on Arduino," *Procedia Computer Science*, vol. 166, pp. 274-278, 2020.

[8] Y. Jin, S. Li, J. Li, H. Sun and Y. Wu, "Design of an Intelligent Active Obstacle Avoidance Car Based on Rotating Ultrasonic Sensors,"

2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), pp. 753-757, 2018.

[9] S. K. Satti, K. Suganya Devi, P. Dhar and P. Srinivasan, "A machine learning approach for detecting and tracking road boundary lanes," *ICT Express*, vol. 7, no.1, pp. 99-103, 2021.

[10] X. Wang, L. Xu, H. Sun, J. Xin and N. Zheng, "On-Road Vehicle Detection and Tracking Using MMW Radar and Monovision Fusion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no.7, pp. 2075-2084, 2016.

[11] Z. Nie and J. Qu, "Multi-task Autonomous Driving Based on Improved Convolutional Neural Network and ST Loss in MTS and MOD Modes," *Current Applied Science and Technology*, vol. 23, no.3, pp. 10-36, 2023.

[12] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent and J. Weaver, "Google Street View: Capturing the World at Street Level," *Computer*, vol. 43, No.6, pp. 32-38, 2010.

[13] J. QU, "Environments of Automatic Driving," class lecture. Selected Topic in Computer and Information Technology 1, ET61714, Faculty of Engineering and Technology, Panyapiwat Institute of Management, 2023.

[14] J. Qu, "Tasks for Automatic Driving," class lecture. Selected Topic in Computer and Information Technology 1, ET61714, Faculty of Engineering and Technology, Panyapiwat Institute of Management, 2023.

[15] X. Wang, T. Jiang, and Y. Xie, "A Method of Traffic Light Status Recognition Based on Deep Learning," *Proceedings of the 2018 International Conference on Robotics, Control and Automation Engineering*, pp. 166-170, 2018.

[16] R. R. O. Al-Nima, T. Han and T. Chen, "Road Tracking Using Deep Reinforcement Learning for Self-driving Car Applications," *International Conference on Computer Recognition Systems*, pp. 106-116, 2020.

[17] Y. Li and J. Qu, "Intelligent Road Tracking and Real-time Acceleration-deceleration for Autonomous Driving Using Modified Convolutional Neural Networks," *Current Applied Science and Technology*, vol. 22, no.6, pp. 10-55003, 2022.

[18] H. Jian, "Design of automatic obstacle Avoidance Car Based on STM32," *Proceedings of the 2017 6th International Conference on Measurement, Instrumentation and Automation (ICMIA 2017)*, pp. 311-314, 2017.

[19] C. Zhou, F. Li, W. Cao, C. Wang, and Y. Wu, "Design and implementation of a novel obstacle avoidance scheme based on a combination of CNN-based deep learning method and liDAR-based image processing approach," *Journal of Intelligent & Fuzzy Systems*, vol. 35, no.2, pp. 1695-1705, 2018.

[20] H.-Y. Lin, C.-C. Chang, V. L. Tran, and J.-H. Shi, "Improved traffic sign recognition for in-car cameras," *Journal of the Chinese Institute of Engineers*, vol. 43, no.3, pp. 300-307, 2020.

[21] J. Han, X. Liang, H. Xu, K. Chen, L. Hong, C. Ye, W. Zhang, Z. Li, X. Liang, and C. Xu, "Soda10m: Towards Large-Scale Object Detection Benchmark for Autonomous Driving," https://arxiv.org/abs/2106.11118, 2023.

[22] S. Ding and J. Qu, "A Study on Safety Driving of Intelligent Vehicles Based on Attention Mechanisms," *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, vol. 16, no.4, pp. 410-421, 2022.

[23] P. Sun, H. Kretzschmar, X. Dotiwalla, A. e. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen and D. Anguelov, "Scalability in Perception for Autonomous Driving Waymo Open Dataset," *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2446-2454, 2020.

[24] Z. Bao, S. Hossain, H. Lang and X. Lin, "High-Definition Map Generation Technologies For Autonomous Driving," [Online]. Available: `https://arxiv.org/abs/2206.05400`, 2022.

[25] Y. Li and J. Qu, "MFPE: A Loss Function based on Multi-task Autonomous Driving," ECTI Transactions on Computer and Information Technology (ECTI-CIT), vol. 16, no.4, pp. 393-409, 2022.

[26] J. Nine and R. Mathavan, "Traffic Light and Back-light Recognition using Deep Learning and Image Processing with Raspberry Pi," Embedded Selforganising Systems, vol. 8, no.2, pp. 15-19, 2021.

[27] D. Wang, X. Ma, and X. Yang, "TL-GAN Improving Traffic Light Recognition via Data Synthesis for Autonomous Driving," [Online]. Available: `https://arxiv.org/abs/2203.15006`, 2022.

[28] S. Bali, T. Kumar, and S. S. Tyagi, "Development and performance evaluation of object and traffic light recognition model by way of deep learning," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 26, no.3, pp. 1486-1494, 2022.

[29] R. Niroumand, L. Hajibabai, and A. Hajbabaie, "White Phase Intersection Control through Distributed Coordination: A Mobile Controller Paradigm in a Mixed Traffic Stream," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no.3, pp. 2993-3007, 2023.

[30] S. Ding and J. Qu, "Research on Multi-tasking

Smart Cars Based on Autonomous Driving Systems," *SN Computer Science*, vol. 4, no.3, pp. 292-308, 2023.

[31] S. Riyana and N. Riyana, "Achieving Anonymization Constraints in High-Dimensional Data Publishing Based on Local and Global Data Suppressions," *SN Computer Science*, vol. 3, no.1, pp. 1-12, 2021.

[32] S. Riyana, "Achieving Anatomization Constraints in Dynamic Datasets," *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, vol. 17, no.1, pp. 27-45, 2023.

[33] Mehmet Ercan Nergiz, Maurizio Atzori and Y. Saygin, "Towards Trajectory Anonymization: a Generalization-Based Approach," *Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS*, pp. 52-61, 2008.

[34] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778, 2016.

[35] Q. Hou, D. Zhou, and J. Feng, "Coordinate Attention for Efficient Mobile Network Design," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13713-13722, 2021.

[36] D. W. Scott, "Sturges' rule," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 1, no.3, pp. 303-306, 2009.

Shang Shi is currently studying for the Master of Engineering Technology, Faculty of Engineering and Technology, Panyapiwat Institute of Management, Thailand. He received E.E. from Nanjing Tech University Pujiang Institute, China, in 2022. His research interests are Research direction is artificial intelligence, image processing, and autonomous driving.

Jian Qu is an Assistant Professor at the Faculty of Engineering and Technology, Panyapiwat Institute of Management. He received Ph.D. with Outstanding Performance award from Japan Advanced Institute of Science and Technology, Japan, in 2013. He received B.B.A with Summa Cum Laude honors from Institute of International Studies of Ramkhamhaeng University, Thailand, in 2006, and M.S.I.T from Sirindhorn International Institute of Technology, Thammast University, Thailand, in 2010. He has been severing as a house committee for Thai SUPERAI project since 2020. His research interests are natural language processing, intelligent algorithms, machine learning, machine translation, information retrieval, image processing, and autonomous driving.