Deep Learning-Based Lane-Keeping Assist System for Self-Driving Cars Using Transfer Learning and Fine Tuning

Phuc Phan Hong, Huy Hua Khanh, Nghi Nguyen Vinh, Nguyen Nguyen Trung, Anh Nguyen Quoc, and Hoang Tran Ngoc *

Software Engineering Department, FPT University, Can Tho 94000, Vietnam
Email: PhucPHCE171166@fpt.edu.vn (P.P.H.); Huyhkce171778@fpt.edu.vn (H.H.K);
Nghinvce182108@fpt.edu.vn (N.N.V); Nguyenntce182351@fpt.edu.vn (N.N.T);
AnhNQCE170483@fpt.edu.vn (A.N.Q.); Hoang2531992@gmail.com (H.T.N.)
*Corresponding author

***Abstract*—This paper presents an advanced lane-keeping assistance system specifically designed for self-driving cars. The proposed model combines the powerful Xception network with transfer learning and fine-tuning techniques to accurately predict the steering angle. By analyzing camera-captured images, the model effectively learns from human driving knowledge and provides precise estimations of the steering angle necessary for safe lane-keeping. The transfer learning technique allows the model to leverage the extensive knowledge acquired from the ImageNet dataset, while the fine-tuning technique is utilized to tailor the pre-trained model to the specific task of steering angle prediction based on input images, enabling optimal performance. Fine-tuning was initiated by initially freezing the pre-trained model and training only the Fully Connected (FC) layer for the first 10 epochs. Subsequently, the entire model, encompassing both the backbone and the FC layer, was unfrozen for further training. To evaluate the system's effectiveness, a comprehensive comparative analysis is conducted against popular existing models, including Nvidia, MobilenetV2, VGG19, and InceptionV3. The evaluation includes an assessment of the operational accuracy based on the loss function, specifically utilizing the Mean Squared Error (MSE) equation. The proposed model achieves the lowest loss function values for both training and validation, demonstrating its superior predictive performance. Additionally, the model's performance is further evaluated through extensive real-world testing on pre-designed trajectories and maps, resulting in the minimal deviation of the steering angle from the desired trajectory over time. This practical evaluation provides valuable insights into the mode's reliability and its potential to effectively assist in lane-keeping tasks.**

***Keywords*—lane-keeping assistance, autonomous vehicles, Xception, transfer learning, fine-tuning, steering angle prediction**

I. INTRODUCTION

In recent years, there has been a growing interest in the research field of self-driving autonomous cars and vehicles, which falls under the broader domain of artificial intelligence. This field is rapidly evolving and holds immense potential to transform the automotive industry. Self-driving technology offers the promise of safer driving practices by enabling vehicles to operate in an automated manner, thus reducing the frequency of road accidents and minimizing economic losses. It provides a safer alternative to human drivers, capable of functioning autonomously without the need for human intervention. Autonomous driving encompasses six distinct levels of automation, ranging from zero to full automation, that define the degree of autonomy and human involvement in the driving process. Initially, rule-based strategies dominated autonomous driving research with a strong emphasis on image processing. These approaches treated perception and control as separate modules, operating independently [1–10]. However, the emergence of deep learning has led to a significant shift towards end-to-end vehicle control as a prominent research area in autonomous driving [11, 12]. This approach seamlessly integrates perception and control, leveraging the power of deep learning to optimize the performance of autonomous driving systems.

Several notable studies have contributed to the advancement of autonomous driving technology. Li *et al.* [13] presented an active traffic lane management method for intelligent connected vehicles, aiming to enhance urban driving by improving traffic capacity, lane management, and alleviating congestion on freeways. Recently, Nvidia introduced a pioneering end-to-end driving model for steering angle control [14]. This model utilizes Convolutional Neural Networks (CNN) to directly predict the steering angle by analyzing raw pixel data from a single frame captured by a front-view camera. Subsequent research studies have explored various CNN architectures. Simmons *et al.* [15] proposed a lane-following model based on a combination of Deep Neural Networks (DNN) and CNN to enhance accuracy and minimize loss. The LeNet network architecture, comprising activation functions, convolutional layers, and fully connected layers, was employed in other research papers [16, 17] to maximize

Manuscript received September 15, 2023; revised October 11, 2023; accepted October 25, 2023; published March 8, 2024.

doi: 10.12720/jait.15.3.322-329

performance and achieve optimal results. Gupta *et al.* [18] employed the MobileNetV2 model for lane-keeping, aiming to minimize the mean square error between the true and predicted steering angles. Babiker *et al.* [19] conducted an experiment combining the VGG-19 model for classification with an end-to-end steering control system to simultaneously detect traffic signals and steering angles, achieving a signal accuracy score of 86%. In 2015, Google unveiled the InceptionV3 model, a highly acclaimed architecture renowned for its exceptional performance in image classification tasks [20]. One of its distinctive features is the incorporation of the "Inception module", which employs parallel convolutions with varying filter sizes, allowing the network to capture information at multiple scales [21]. This innovative design has found successful applications in various domains, including lane-keeping systems for autonomous vehicles by Park *et al.* [22]. However, these methods suffer from several limitations, including suboptimal coefficient consideration and low accuracy due to the model's structure not being fully optimized.

In order to overcome the limitations of existing methods in terms of accuracy, we propose the utilization of the Xception model as an improvement over InceptionV3. Our research focuses on enhancing the lane-keeping system for autonomous vehicles by employing various techniques. These techniques include data augmentation, transfer learning, fine-tuning, and the redesign of the fully connected layers. Our approach involves taking images captured by a camera as input and predicting the steering angle with optimized accuracy. The Xception model, which builds upon the foundation of InceptionV3, incorporates depthwise separable convolutions. This innovative technique separates spatial and channel-wise convolutions, reducing computational complexity while maintaining high performance [23]. By integrating the Xception architecture into our system, we aim to achieve precise and reliable predictions of the steering angle. To improve generalization to real-world scenarios, we employ data diversity techniques to augment our training dataset. Additionally, transfer learning is applied by leveraging pre-trained weights from the Xception model, which has been trained on a large-scale image dataset [24]. We further fine-tune the model specifically for the lane-keeping task, adapting it to the unique requirements of autonomous driving. To tailor the output specifically to the prediction of steering angles, we redesign the fully connected layers of the Xception model. This customization allows us to optimize the accuracy of the model's predictions, enabling precise control of the vehicle within the lane boundaries. To evaluate the accuracy of the proposed algorithm, we constructed a vehicle model and a corresponding road map with lanes. We will compare the accuracy based on the loss function, which measures the deviation between the predicted and actual values, as well as the error of the overall trajectory followed by the vehicle.

The remaining sections of this article are organized as follows. In Section II, we provide an overview of the proposed method. Section III focuses on the methodologies, including the models used for transfer learning, the feature extraction models, and the fine-tuning approach. In Section IV, we present the experimental system, dataset, evaluation metrics, and results, along with detailed discussions. Finally, in Section V, we provide a comprehensive conclusion summarizing the findings and discussing their implications.

II. Overview of the Proposed Method

The proposed system consists of a front-facing RGB camera and a robust Xception model that estimates the steering wheel angle based on the camera input. Fig. 1 illustrates a simplified block diagram of the proposed method, which involves four main steps.

- Step 1-(Transfer Learning for Feature Extraction): The model utilizes a transfer learning technique by leveraging pre-trained weights from a large Imagenet dataset to ensure optimal feature extraction capabilities.
- Step 2-(Redesigning the Fully Connected (FC) Layers): The Fully Connected (FC) layers of the model are redesigned with an appropriate activation function to produce accurate predictions of the steering angle for lane-keeping.
- Step 3-(Training with Frozen Feature Extraction Models): Training is performed with the feature extraction models frozen to find the optimal parameters for the fully connected layers.
- Step 4-(Unfreezing the Feature Extraction Model): The feature extraction model is unfrozen, and the entire system is trained to find the best overall parameters for the system.

By following these four steps, the proposed system systematically optimizes the model and enhances the accuracy of the steering angle prediction, making it well-suited for lane-keeping in self-driving cars.

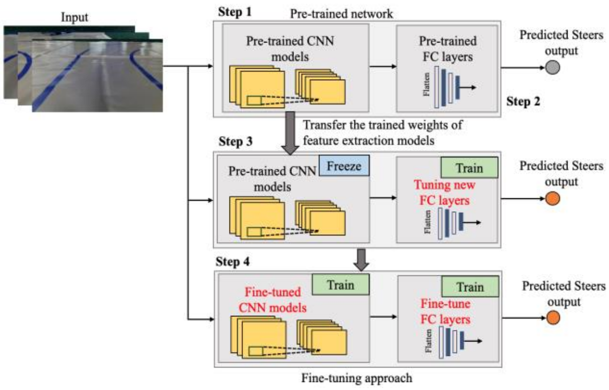

Fig. 1. Block diagram of the proposed method.

III. Methodologies

A. Utilizing Pre-trained Models for Transfer Learning

Transfer learning offers a promising solution for deep learning networks with limited training data and high costs associated with labeled data creation [25]. It involves two main strategies: feature extraction and fine-tuning. In feature extraction, a pre-trained model is used to extract activation values from different layers, accelerating the training process by reusing pre-existing parameters. By leveraging knowledge from large-scale image datasets such as Imagenet [26], Microsoft COCO [27], Open Images dataset [28], and Places365 [29], transfer learning enhances the prediction of steering angles in autonomous driving tasks. It overcomes the scarcity of labeled driving data and achieves high accuracy.

This technique significantly improves the efficiency and precision of autonomous driving systems, benefiting the advancement of the field and contributing to safer transportation.

B. Extracting Features Models

Steering angle prediction is a critical task in autonomous driving systems, as it enables vehicles to navigate accurately and safely. Convolutional Neural Networks (CNNs) have proven to be effective in extracting meaningful features from input images (size 180×180×3) and predicting the corresponding steering angles (range 0°–180°). In this study, we explore the utilization of various CNN models for steering angle prediction and analyze their performance and capabilities.

1) Nvidia-CNN model

The Nvidia model is specifically designed for steering angle prediction [14]. It employs a series of convolutional layers, including 3 layers with a kernel size of 5×5 and 2 layers with a kernel size of 3×3. The output is then fed into a prediction layer to estimate the steering angle. The model utilizes the Exponential Linear Unit (ELU) activation function, which is well-suited for the task at hand. The structure of the steering angle prediction model based on the Nvidia-CNN architecture is shown in Fig. 2(a).

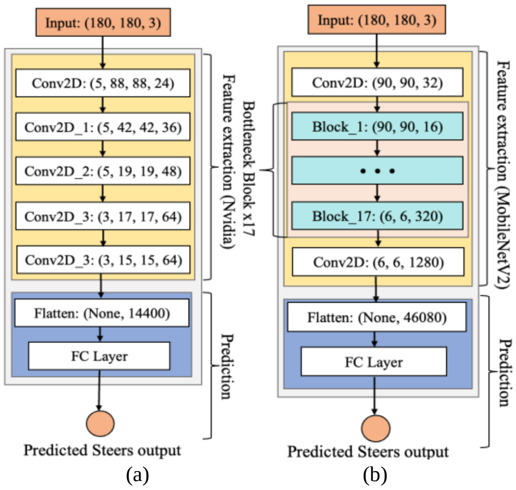

Fig. 2. Structure of the steering angle prediction model: (a) Based on the Nvidia-CNN architecture; (b) Based on the MobileNetV2 architecture.

2) MobileNetV2 model

MobileNetV2 utilizes an inverted residual structure and linear bottleneck to create an efficient layer structure. It employs separable depth-wise convolutions, which split the standard convolution into two separate functions [30]. This approach reduces computational requirements and the overall model size, as depicted in Fig. 2(b). The MobileNetV2 architecture includes 17 bottleneck blocks that further enhance its performance. These blocks leverage the power of residual connections and depth-wise point-wise convolutions to optimize accuracy while minimizing computational costs. Additionally, MobileNetV2 is specifically designed for low-latency image processing, making it an ideal choice for hardware-constrained devices.

3) VGG19 model

The VGG networks, including the renowned AlexNet [31], follow a classic architecture consisting of convolutional, max-pooling, and activation layers, followed by fully-connected classification layers. The VGG-19 CNN network model [32] is a notable architectural variation. By incorporating VGG-19 with additional output layers, precise steering angle prediction can be achieved. The model extends the original VGG-16 architecture by incorporating three additional convolutional layers, resulting in a significant enhancement in steering angle estimation accuracy. The architecture of the steering angle prediction model based on VGG-19 is illustrated in Fig. 3(a).

4) InceptionV3 model

The InceptionV3 model, initially proposed by Szegedy *et al.* [33], was specifically designed to handle the ImageNet dataset [26]. It comprises 11 inception models stacked together, combined with global average pooling, to effectively learn multi-level features for image classification. Each inception model, as depicted in Fig. 3(b), employs 1×1, 3×3, and 5×5 convolutions to extract features from various spatial sizes. The utilization of 1×1 convolutions helps reduce the volume size, enabling efficient information processing.

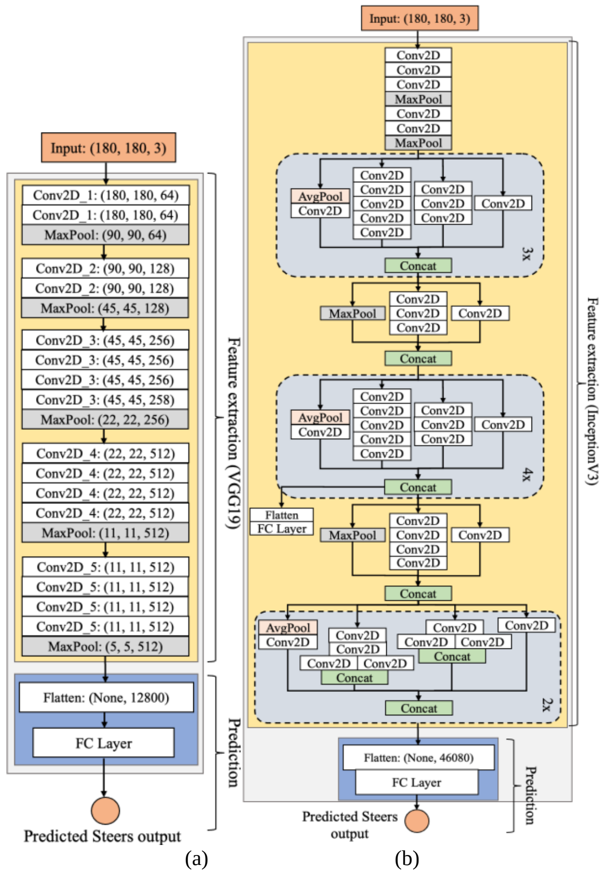

Fig. 3. Structure of the steering angle prediction model: (a) Based on the VGG19 architecture; (b) Based on the InceptionV3 architecture.

By stacking these Inception modules, the model is capable of capturing intricate hierarchical features. InceptionV3 further utilizes maximum and average pooling operations to reduce the dimensionality of the feature maps, aiding in efficient computation and extraction of important characteristics.

5) Xception model

Although the mentioned methods have been effectively applied in steering angle prediction, the accuracy still remains limited. Therefore, we have utilized a new model called Xception [34], in conjunction with transfer learning and fine-tuning techniques, to achieve optimal parameter settings for the system.

- The separable convolution in Xception, also known as extreme inception, is a variation of the original depthwise separable convolution. It's essentially a sequence of depthwise separable convolution layers with residual connections but with a reversed order compared to the original depthwise separable convolution. This modified separable convolution is aimed at reducing the number of computations in conventional convolution models, making it more cost-efficient [35]. While reducing parameters might have potential trade-offs, when used appropriately, it offers efficiency without compromising model performance. These stacked separable convolutions are treated as an inception module within the Xception architecture. Fig. 4 shows the diagram of an example modified depthwise separable convolution. The formulas provided below depict the size and number of filters involved in this convolution:

$$\begin{array}{lll} Input: & H_P \times W_P \times 4 & \\ Conv: & 1 \times 1 \times N & \text{with } N = 3 \text{ filters} \\ Output: & H_P \times W_P \times N & \\ Conv: & 3 \times 3 \times M & \text{with M = 3 filters} \\ Output: & H_D \times W_D \times M & \end{array} \quad (1)$$

where $H_P \times W_P$ are the height and width of the input image, $H_D \times W_D$ are the height and width of the output image, N is the number of pointwise filters, and M is the number of depthwise filters.

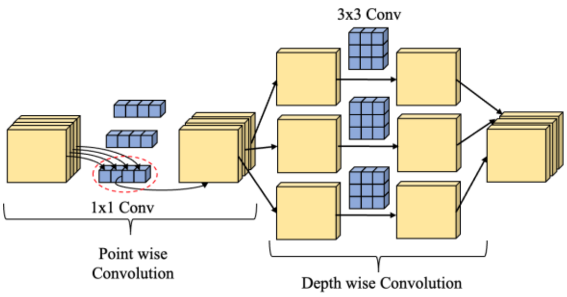

Fig. 4. Diagram showcases the depthwise separable convolution employed in the Xception network.

- Xception architecture consists of three main sections: the entry flow, middle flow, and exit flow. In each of these flows, both conventional and separable convolutions are used, and they are followed by Batch Normalization to normalize the convolutional layer outputs, improving model stability and performance. By integrating the Xception architecture into our steering angle prediction system, we aim to boost prediction accuracy and reliability, thus advancing the safety and efficiency of autonomous driving systems. You can see the model's structure based on the Xception architecture in Fig. 5.

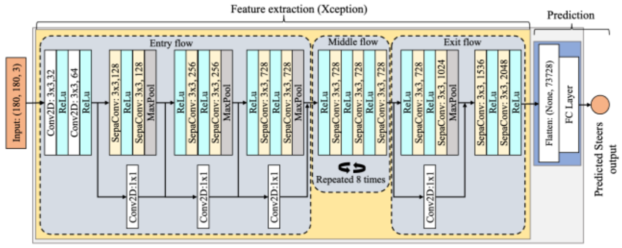

Fig. 5. Structure of the steering angle prediction model based on the Xception architecture.

C. Fine-Tuning Approach

The fine-tuning approach was employed after utilizing the transfer learning technique by incorporating pre-trained extracting features models from the Imagenet dataset into the base model. In the initial 10 training iterations, the base models were frozen, and only the weights of the Prediction layer, including Flatten and FC layers, were updated. These weights were saved when the Loss function value reached a lower value, indicating higher accuracy. After completing the initial 10 training iterations, the base models were unfrozen, and an additional 20 iterations of training were performed. During this stage, all the weights in the extracting features models were updated and optimized. To ensure a fair comparison, the training and evaluation datasets were consistent, and all the models, including Nvidia-CNN, MobileNetV2, VGG19, and InceptionV3, underwent the same fine-tuning process as the proposed Xception model. Following the Flatten layer, the extracted features were passed through four fully connected layers with 2048, 1000, 100, and 1 units, respectively, each with ELU activation for the predicted steering angle output.

IV. Experimental Results

A. Experimental System

A Donkey Car, built using a Raspberry Pi, is utilized to gather data and assess the proposed models. The car consists of three primary components: a Raspberry Pi 4-8 GB RAM for computational processing, a Creative BlasterX® Senz3D® camera for image capture, and a servo motor for steering angle control. Fig. 6 depicts the lane-keeping map, block diagram of the proposed method, and the overall system. Image data from the camera and steering angle inputs from the joystick are collected and used for training on a computer system equipped with Ubuntu 20.04, an Intel Core I3-10105F CPU, an Nvidia GTX 1660-8 GB GPU, and 16 GB RAM. The implemented algorithms are written in Python 3.10 and employ the Tensorflow 2.12.0 and Keras 2.12.0 libraries. During the training and comparison processes, all networks are trained using the Adam optimizer with a learning rate of 0.001. The activation function used for the system's output layer, which estimates the steering angle, is the Exponential Linear Unit (ELU).

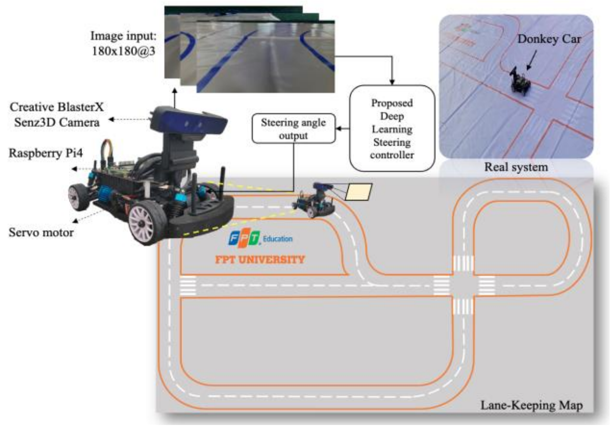

Fig. 6. Experimental self-driving car and block diagram of the proposed system.

B. Dataset and Evaluation Metrics

1) Dataset

The Donkey self-driving car is designed and utilized in two different modes: data collection and autonomous driving.

During the data collection mode, we utilize a driving wheel joystick to control the car and ensure it stays within the lane. The car has a steering angle limit of 0 to 180 degrees. Additionally, the car is equipped with a front-facing camera that captures images. The controller records both the steering angle input from the joystick and the corresponding images from the camera. These data are collected at a rate of 30 frames per second and used as training data. In the autonomous driving mode, the car is equipped with the trained models and evaluated for accuracy and performance during self-driving. The dataset used in this study consists of 20,000 images, which are continuously collected along with their corresponding steering angles determined by human perception. The distribution of the dataset and example input images for training can be seen in Figs. 7 and 8.

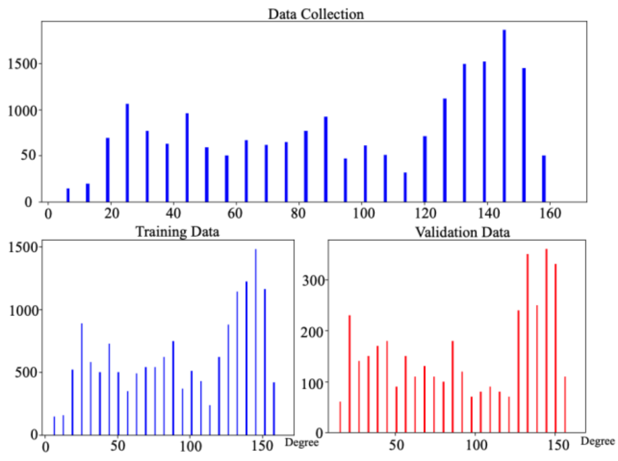

Fig. 7. Distribution of the dataset for training.

2) Evaluation metrics

In order to achieve accurate steering angle prediction, we approached it as a regression problem. The evaluation of the deviation between the predicted steering angle and the actual steering angle is done using the Mean Square Error (MSE) as the loss function Eq. (2). By squaring the differences, MSE effectively captures any variations between the predicted and actual steering angles, making it a robust metric for assessment.

$$MSE(Y, \hat{Y}) = \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)^2 \tag{2}$$

where Y_i and $\hat{Y}_i$ are the true steering angles and predicted value, n is the number of data points.

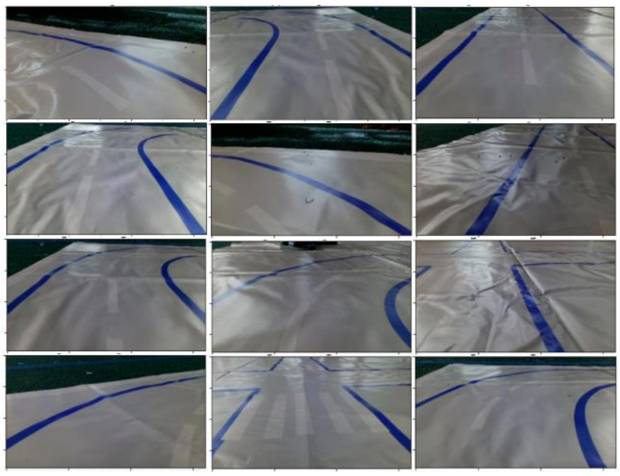

Fig. 8. Some example input images of the dataset.

C. Results

1) Training results comparison

The training results were compared among different models before and after applying transfer learning and fine-tuning techniques. This comparison can be observed in Fig. 9 and summarized in Tables I and II.

During the initial 10 epochs, the minimum values of training loss and validation loss for Nvidia-CNN, MobileNetV2, VGG19, InceptionV3, and Xception models were as follows: (2117.2 & 2163.9), (1085.7 & 1048.1), (880.7 & 513.2), (918.4 & 676.2), and (597.3 & 491.0), respectively. The error values decreased gradually, with the Xception model exhibiting the lowest Mean Squared Error (MSE) between the predicted and truth steering angles. However, the remaining error was still significant, which could affect the lane-keeping performance and the overall safety of the self-driving car. To address this, we applied fine-tuning for the next 20 epochs. The training loss and validation loss values for each model were as follows: Nvidia-CNN (1629.2 & 1448.1), MobileNetV2 (934.5 & 729.4), VGG19 (744.5 & 419.3), InceptionV3 (567.2 & 484.4), and the proposed Xception-based method (265.2 & 386.8).

After applying transfer learning and fine-tuning, the model coefficients were optimized and tuned specifically for the steering angle output. The error of validation loss was significantly improved on the proposed model (reduced from 491.0 to 386.8). These results indicate that the proposed method outperformed the other approaches. To provide a more objective evaluation, we implemented the models into the lane-keeping system, as shown in Fig. 6, for further testing and assessment.

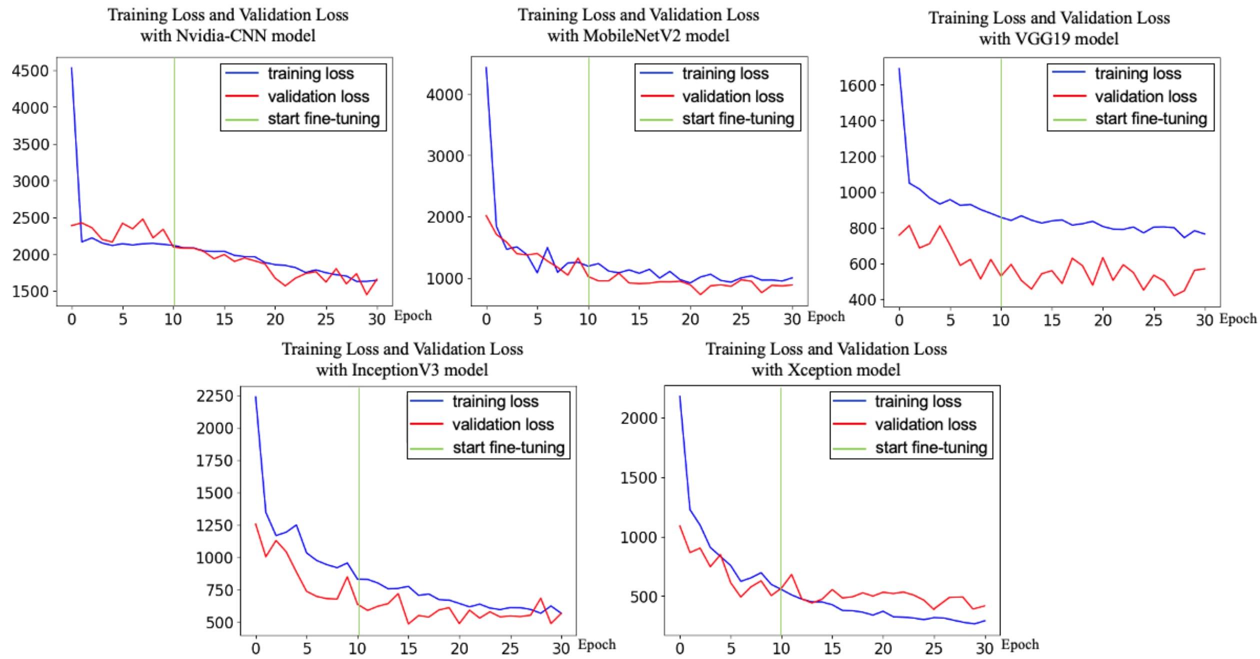

Fig. 9. Training loss and validation loss results of the Nvidia-CNN, MobileNetV2, VGG19, InceptionV3, and Xception models before and after fine-tuning.

TABLE I. THE MINIMUM VALUE OF TRAINING LOSS

Approach	Nvidia-CNN	MobileNet V2	VGG 19	Inception V3	Xception
10 Epochs	2117.2	1085.7	880.7	918.4	597.3
20 Epochs (F-tuning)	1629.2	934.5	744.5	567.2	265.2

TABLE II. THE MINIMUM VALUE OF VALIDATION LOSS

Approach	Nvidia-CNN	MobileNet V2	VGG 19	Inception V3	Xception
10 Epochs	2163.9	1048.1	513.2	676.2	491.0
20 Epochs (F-tuning)	1448.1	729.4	419.3	484.4	386.8

2) *Testing results comparison*

The implemented models were tested on a Raspberry Pi 4 using the Donkey Car platform. The car was controlled using a joystick to keep it within the lane, and the steering angle data was recorded. Subsequently, the car was set to autonomous driving mode, and the predicted steering angles were recorded as it followed the same trajectory as the joystick-controlled run.

The comparison between the human-perceived steering angle from the joystick and the predicted steering angle from the models is shown in Figs. 10 and 11. Clearly, the proposed model accurately predicts the steering angle and closely follows the ground truth angle of the actual road trajectory. We can examine the details in the broken green circle. It's easy to see that the proposed method (A-5) with the blue "predicted angle" line closely aligns with the red "ground truth angle" line. In contrast, the predicted and ground truth angles for the Nvidia method (A-1) exhibit the largest deviations. It can be concluded that the proposed method significantly improves accuracy, resulting in better lane-keeping performance compared to other models.

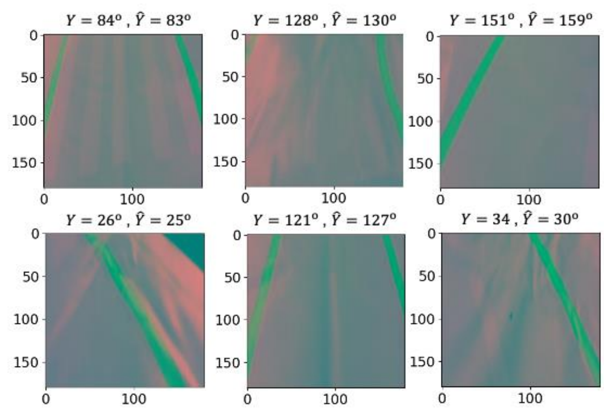

Fig. 10. Examples of steering angle predictions made by the proposed model compared to the ground truth steering angle.

V. CONCLUSION

This study introduces an advanced lane-keeping assistance system for autonomous vehicles. The proposed model combines the Xception network with transfer learning and fine-tuning methods to accurately predict the steering angle using camera images. Through extensive comparisons with existing models, such as Nvidia, MobileNetV2, VGG19, and InceptionV3, our model demonstrates superior performance with significantly lower loss values during training and validation. Real-world testing on predefined trajectories and maps further validates the reliability and effectiveness of our model in maintaining precise and reliable lane-keeping.

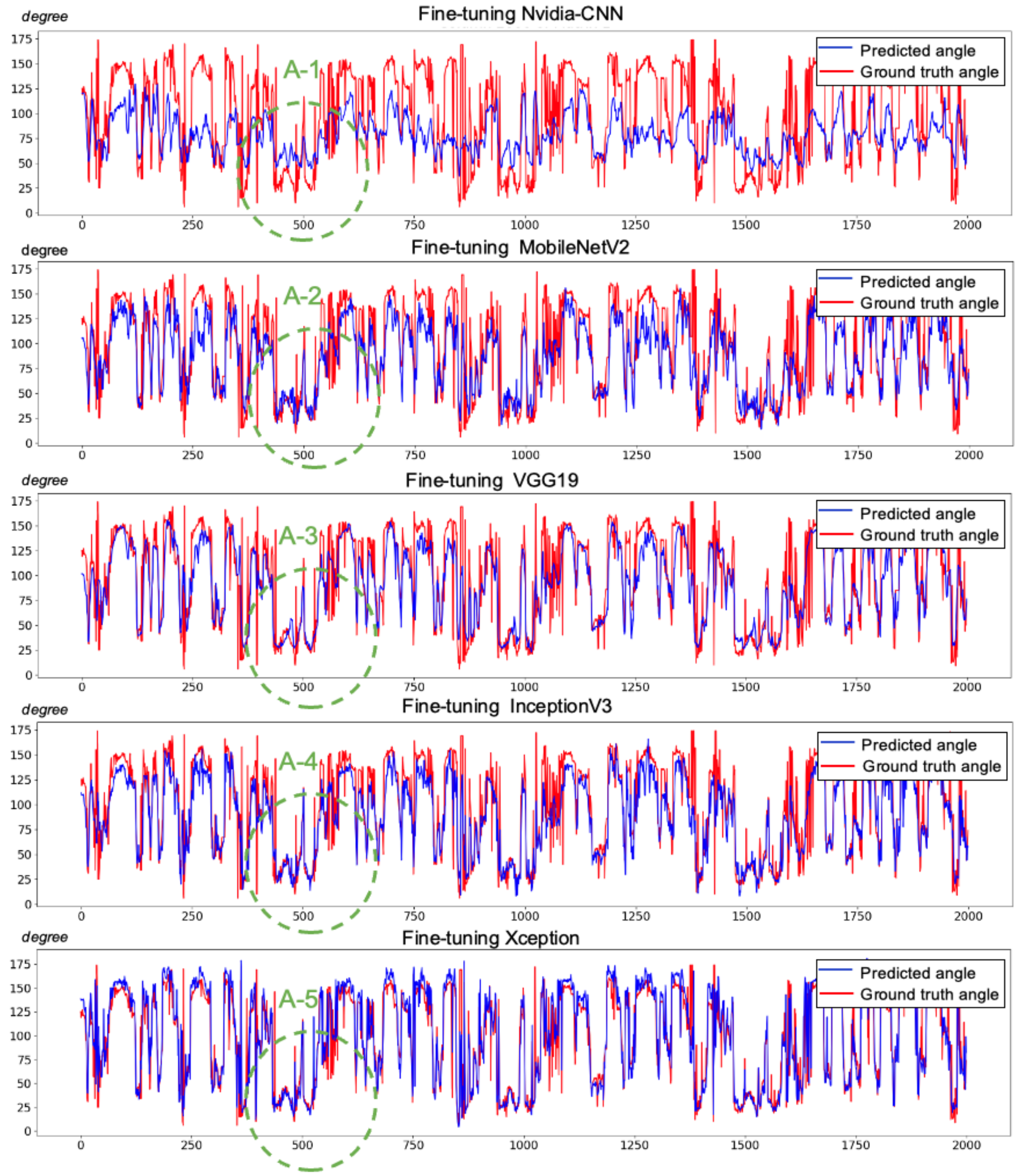

Fig. 11. Comparison between the human-perceived steering angle from the joystick and the predicted steering angle from the models.

Future research can explore the integration of image sequence processing techniques to capture temporal dynamics, investigate advanced deep learning architectures tailored for sequential data modeling, and incorporate additional sensor modalities to enhance perception. Comprehensive evaluations in diverse real-world driving scenarios will strengthen the model's robustness and effectiveness.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Hoang Tran Ngoc had the idea for the research and provided support for Phuc Phan Hong and Anh Nguyen Quoc. Phuc Phan Hong and Huy Hua Khanh conducted the research, while Anh Nguyen Quoc, Nghi Nguyen Vinh, and Nguyen Nguyen Trung wrote the paper and prepared the dataset; all authors approved the final version.

References

[1] H. N. Tran and L. Quach, "Adaptive lane keeping assist for an autonomous vehicle based on steering fuzzy-PID control in ROS," *International Journal of Advanced Computer Science and Applications*, vol. 13, issue 10, 2022. doi: 10.14569/IJACSA.2022.0131086

[2] P. H. Phan, A. Q. Nguyen, L. Quach, and H. N. Tran, "Robust autonomous driving control using auto-encoder and end-to-end deep learning under rainy conditions," in *Proc. the 2023 8th International Conference on Intelligent Information Technology (ICIIT'23)*, New York, USA, 2023, pp. 271–278. https://doi.org/10.1145/3591569.3591616

[3] H. K. Hua, K. H. Nguyen, L. Quach, and H. N. Tran, "Traffic lights detection and recognition method using deep learning with improved YOLOv5 for autonomous vehicle in ROS2,". in *Proc. the 2023 8th International Conference on Intelligent Information Technology (ICIIT'23)*, New York, NY, USA, 2023, pp. 117–122. https://doi.org/10.1145/3591569.3591589

[4] J. Janai, F. Güney, A. Behl, and A. Geiger, "Computer vision for autonomous vehicles: Problems, datasets and state of the art," arXiv preprint, arXiv:1704.05519, 2017.
[5] V. D. Nguyen, T. D. Trinh, and H. N. Tran, "A robust triangular sigmoid pattern-based obstacle detection algorithm in resource-limited devices," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 6, pp. 5936–5945, June 2023. doi: 10.1109/TITS.2023.3253509
[6] H.-T. Vo, H. N. Tran, and L. Quach, "An approach to hyperparameter tuning in transfer learning for driver drowsiness detection based on bayesian optimization and random search," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 14, no. 4, 2023. doi: 10.14569/IJACSA.2023.0140492
[7] H. N. Tran, H. V. N. Nguyen, K. H. Nguyen, and L.-D. Quach, "Lane road segmentation based on improved unet architecture for autonomous driving," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 14, no. 7, 2023. http://dx.doi.org/10.14569/IJACSA.2023.0140724
[8] H. N. Tran, K. H. Nguyen, H. K. Hua, H. V. N. Nguyen, and L.-D. Quach, "Optimizing YOLO performance for traffic light detection and end-to-end steering control for autonomous vehicles in gazebo-ROS2," *International Journal of Advanced Computer Science and Applications(IJACSA)*, vol. 14, no. 7, 2023.
[9] T. T. Khanh, T. Hoang Hai, V. Nguyen, T. D. T. Nguyen, N. Thien Thu, and E.-N. Huh, "The practice of cloud-based navigation system for indoor robot," in *Proc. 2020 14th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, Taichung, Taiwan, 2020, pp. 1–4. doi: 10.1109/IMCOM48794.2020.9001709
[10] N. T. Thu, M. A. Layek, and E. N. Huh, "An effective vision-based self-navigation system for autonomous indoor vehicle," *KIISE Transactions on Computing Practices*, vol. 25, no. 7, pp. 351–356, 2019. doi: 10.5626/KTCP.2019.25.7.351
[11] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: learning affordance for direct perception in autonomous driving," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2722–2730. doi: 10.1109/ICCV.2015.312
[12] M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. Jackel, and U. Muller, "Explaining how a deep neural network trained with end-to-end learning steers a car," arXiv preprint, arXiv:1704.07911, 2017.
[13] H. Li, J. Zhang, Z. Zhang, and Z. Huang, "Active lane management for intelligent connected vehicles in weaving areas of urban expressway," *Journal of Intelligent and Connected Vehicles*, vol. 4, no. 2, pp. 52–67, 2021. doi: 10.1108/JICV-08-2020-0009
[14] M. Bojarski, D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," arXiv preprint, arXiv:1604.07316, 2016.
[15] B. Simmons, P. Adwani, H. Pham, Y. Alhuthaifi, and A. Wolek, "Training a remote-control car to autonomously lane-follow using end-to-end neural networks," in *Proc. 53rd Annual Conference on Information Sciences and Systems (CISS)*, Baltimore, MD, USA, 2019, pp. 1–6. doi: 10.1109/CISS.2019.8692851
[16] V. Rausch, A. Hansen, E. Solowjow, C. Liu, and E. Kreuzer, "Learning a deep neural net policy for end-to-end control of autonomous vehicles," in *Proc. American Control Conf. (ACC)*, Seattle, USA, 2017, pp. 4914–4919. doi: 10.23919/ACC.2017.7963716
[17] R. Hazra, A. Kumar, and B. Baranidharan, "Effect of various activation function on steering angle prediction in CNN based autonomous vehicle system," *Int. Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, pp. 2249–8958, December 2019. doi: 10.35940/ijeat.B4017.129219
[18] M. Gupta, V. Upadhyay, P. Kumar, and F. Al-Tụman, "Deep learning implementation of autonomous driving using ensemble-m in simulated environment," *Research Square*, May 2021. doi: 10.21203/rs.3.rs-483461/v1
[19] M. Babiker, M. Elawad, and A. Ahmed, "Convolutional neural network for a selfdriving car in a virtual environment," in *Proc. 2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, September 2019. doi: 10.1109/ICCCEEE46830.2019.9070826
[20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.
[21] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. of the AAAI Conference on Artificial Intelligence*, 2017, pp. 4278–4284.
[22] M. Park, H. Kim, and S. Park, "A convolutional neural network-based end-to-end self-driving using lidar and camera fusion: Analysis perspectives in a real-world environment," *Electronics*, vol. 10, 2608, 2021. https://doi.org/10.3390/electronics10212608
[23] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. https://doi.org/10.48550/arXiv.1610.02357
[24] O. Russakovsk, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, and L. Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
[25] J. Sokipriala, "Prediction of steering angle for autonomous vehicles using pre-trained neural network," *European Journal of Engineering and Technology Research*, 2021. doi: 10.24018/ejeng.2021.6.5.2537
[26] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. 2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, 2009, pp. 248–255. doi: 10.1109/CVPR.2009.5206848
[27] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," arXiv preprint, arXiv: 1405.0312, 2014. https://doi.org/10.48550/arXiv.1405.0312
[28] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J.-P. Tuset, and V. Ferrari, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *International Journal of Computer Vision*, vol. 126, no. 9, pp. 973–992, 2018. https://doi.org/10.1007/s11263-020-01316-z
[29] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 126, no. 6, pp. 1452–1464, 2017. doi: 10.1109/TPAMI.2017.2723009
[30] W. Li and K. Liu, "Confidence-aware object detection based on MobileNetv2 for autonomous driving," *Sensors*, vol. 21, 2380, 2021. https://doi.org/10.3390/s21072380
[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, 2017. https://doi.org/10.1145/3065386
[32] L. Wen, X. Li, X. Li, and L. Gao, "A new transfer learning based on vgg-19 network for fault diagnosis," in *Proc. 2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, Porto, Portugal, 2019, pp. 205–209. doi: 10.1109/CSCWD.2019.8791884
[33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 2818–2826.
[34] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 1800–1807. doi: 10.1109/CVPR.2017.195
[35] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," arXiv preprint, arXiv:1602.07261, 2016.