End-to-End Steering Angle Prediction for Autonomous Car Using Vision Transformer

Ilvico Sonata[1*], Yaya Heryadi[2], Antoni Wibowo[3], and Widodo Budiharto[4]
[1–3]Computer Science Department, BINUS Graduate Program - Doctor of Computer Science,
Bina Nusantara University
Jakarta 11480, Indonesia
[4]Computer Science Department, School of Computer Science,
Bina Nusantara University
Jakarta 11480, Indonesia
Email: [1]ilvico@binus.ac.id, [2]yayaheryadi@binus.edu, [3]anwibowo@binus.edu,
[4]wbudiharto@binus.edu

***Abstract*—The development of autonomous cars is currently increasing along with the need for safe and comfortable autonomous cars. The development of autonomous cars cannot be separated from the use of deep learning to determine the steering angle of an autonomous car according to the road conditions it faces. In the research, a Vision Transformer (ViT) model is proposed to determine the steering angle based on images taken using a front-facing camera on an autonomous car. The dataset used to train ViT is a public dataset. The dataset is taken from streets around Rancho Palos Verdes and San Pedro, California. The number of images is 45,560, which are labeled with the steering angle value for each image. The proposed model can predict steering angle well. Then, the steering angle prediction results are compared using the same dataset with existing models. The experimental results show that the proposed model has better accuracy regarding the resulting MSE value of 2,991 compared to the CNN-based model of 5,358 and the CNN-LSTM combination model of 4,065. From the results of this experiment, the ViT model can replace the existing model, namely the CNN model and the combination model between CNN and LSTM, in predicting the steering angle of an autonomous car.**

***Index Terms*—Steering Angle Prediction, Autonomous Car, Vision Transformer (ViT)**

I. Introduction

THE development of autonomous cars today cannot be separated from the search for models that can provide safety and comfort for passengers [1, 2]. Autonomous cars are expected to run according to road conditions and traffic ahead. For this reason, the correct prediction of the autonomous car steering angle is essential in presenting a safe and comfortable autonomous car to avoid a collision. Several previous studies in the field of autonomous cars present predictions of steering angle using deep learning models. The most widely used deep learning model is the Convolutional Neural Network (CNN) model [3, 4].

Several previous studies use CNN to predict the steering angle of autonomous cars [5–8]. In their research, they directly predict the steering angle of an autonomous car based on a sequence of raw images captured by the front-facing camera. In their research, CNN is used to extract spatial features and predict the steering angle based on the feature extraction results. A road image dataset with steering angle labels for each image is used to train the CNN model. After the training process, a CNN model obtained can predict the steering angle based on road images taken using the front-facing camera mounted on an autonomous car.

Although CNN is reliable and widely used for image processing and extracting spatial feature tasks, it performs poorly in processing time series data related to temporal feature extraction. The process of detecting and classifying time series data images cannot be carried out properly by the CNN standard model [9, 10]. Several improvements to the CNN model in predicting the steering angle have been made [11–13]. It combines CNN with Long Short-Term Memory (LSTM) to enhance the performance of autonomous car models. Their research uses CNN to extract spatial features and LSTM to extract temporal features. Their results show that the combination of CNN and LSTM improves the accuracy of the steering angle prediction model.

The development of deep learning models to predict the steering angle of autonomous cars does not stop here as deep learning models develop. Moreover, the image data captured by the front-facing camera of an

Received: April 05, 2022; received in revised form: Aug. 31, 2022; accepted: Sept. 01, 2022; available online: Sept. 18, 2023.
*Corresponding Author

autonomous car is a large amount of time series data and updates continuously. These image data are used to predict the steering angle of the autonomous car. The search for more accurate deep learning models and low resources is still carried out. For this reason, the research question can be formulated: What is the most accurate deep learning model that requires low resources to predict the steering angle of an autonomous car?

The Transformer model, a sequence-to-sequence model, was first introduced in 2017 [14]. Initially, the Transformer model is used for Natural Language Processing (NLP), such as machine translation [15, 16] and sentiment analysis [17, 18]. However, the sequence-to-sequence model of this Transformer is very suitable for processing time series data. In its development in 2020, Vision Transformer (ViT) for image processing, such as image detection and classification, was developed [19]. The ViT produces higher accuracy with lower computational resources than the pre-trained CNN ResNet model. Several previous studies using ViT for image processing and classification have been carried out [20–23]. From the previous research results, ViT can be used as a promising alternative method to replace CNN with results that are not inferior to CNN.

From the literature review mentioned, the ViT model has never been used to predict the steering angle of an autonomous car. The research discusses using the ViT to predict steering angle based on road images taken using a camera. Then, the research also compares the results with the state-of-the-art model of the end-to-end steering angle prediction model developed [5, 12].

II. Research Method

The proposed end-to-end model for predicting the steering angle of an autonomous car using ViT can be seen in Fig. 1. The images from the front-facing camera mounted on the autonomous car are extracted for its features related to steering angle prediction, such as the cars in front, the roads, and the guardrails. The steering angle is predicted for each image based on the extraction results. The steering angle prediction result and the corresponding image are displayed sequentially on the simulator to form a video showing the road and steering angle prediction. The predicted value is compared with the actual value to see the accuracy of the prediction results through the resulting Mean Square Error (MSE) value.

The feature extraction process and steering angle prediction are fully carried out by ViT. The ViT architecture, as proposed by previous research [19], can be seen in Fig. 2. Images captured by the autonomous

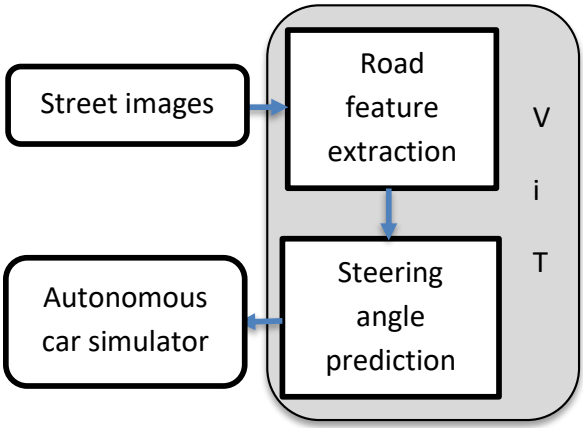


Fig. 1. End-to-end steering angle prediction proposed model.

car's camera are then patched and processed into ViT sequentially.

It is assumed that the street image sequence is $S = \{x_i y_i\}_{i=1}^{n}$, where x is the street view image, and y is the label of each image in the form of steering angle value. The incoming image sequence is patched into several parts. For each x-sized street view image with certain dimensions of $c \times h \times w$, it has c as the number of channels, h as the image height, and w as the width of the image. Image patch processing is carried out with certain dimensions of $c \times p \times p$. It consists of p as the patch size. The patch results are made sequentially through the flattening process. This patch forms a sequence of tokens $(x_1,\ x_2,\ \ldots,\ x_n)$ with length n, where it is $n = \frac{hw}{p2}$. The flattened patch is reduced in size without losing any important features. Then, the reduced flattened patch is converted into a vector sequence by linear embedding, which can be calculated by Eq. (1). It shows E as the learnable embedding matrix, x_{class} as learnable classification token, and E_{pos} as positional information.

$$Z_0 = [x_{class}; x_1E; x_2E; \ldots; x_nE] + Epos, E \in \mathbb{R}^{(p^2.c)xD}, E_{pos} \in \mathbb{R}^{(n+1)xD}. \quad (1)$$

The positional information (E_{pos}) is added to maintain the spatial arrangement of each patch to be the same as the original image. It represents the patch position in the form of a serial number on each patch. The output of the positional embedding is then fed into the standard Transformer Encoder for feature extraction.

The Transformer Encoder consists of Multi-Head Attention (MHA) and Multi-Layer Perceptron (MLP), where it previously places two layers in Layer Normalization (LN), one before MHA and MLP. The LN is used to speed up the training process through statistical

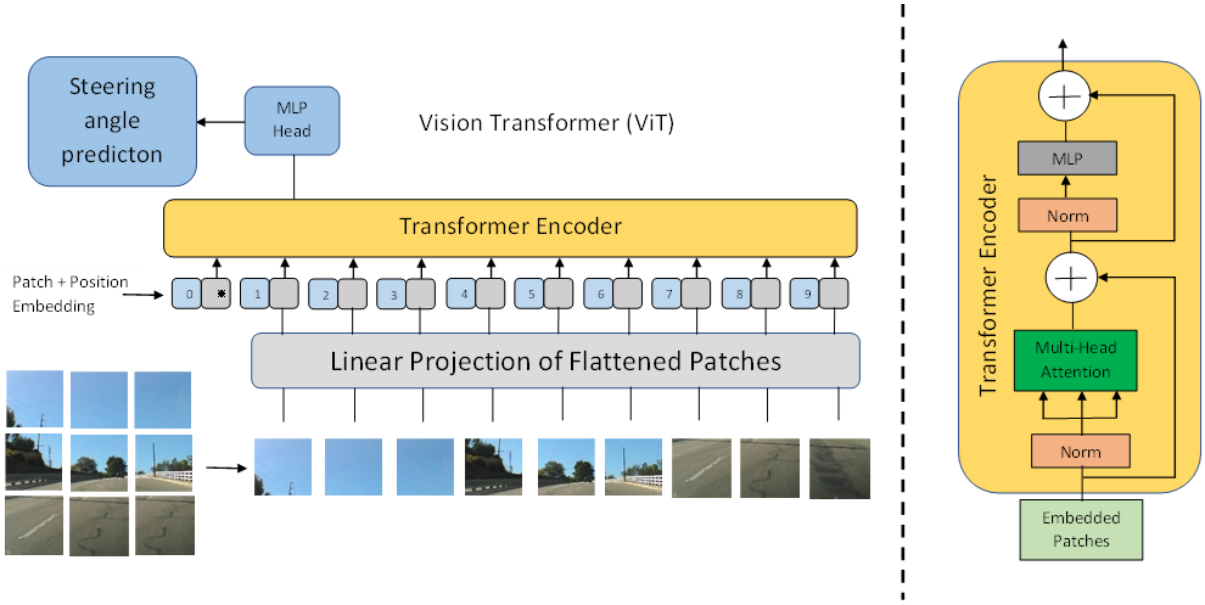


Fig. 2. ViT architecture [19].

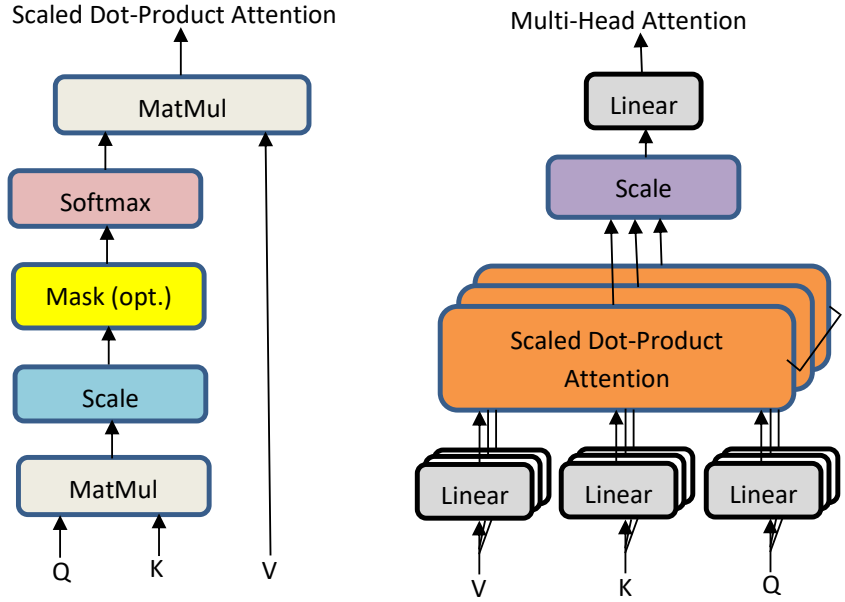


Fig. 3. Attention mechanism [14].

data estimation [24]. The MLP consists of two dense layers and uses the GeLu activation function. The output of MHA can be represented by Eq. (2). Then, the output of MLP can be shown by Eq. (3).

$$z'_\ell = MHA\left(LN\left(z_{\ell-1}\right)\right) + Z_{\ell-1},\ \ell = 1 \ldots L, \quad (2)$$

$$z_\ell = MLP\left(LN\left(z'_\ell\right)\right) + z'_\ell,\ \ell = 1 \ldots L, \quad (3)$$

$$y = LN\left(z_L^0\right). \quad (4)$$

The output of the Transformer Encoder is written in Eq. (4). It has L as the number of identical layers on the Transformer Encoder. The output of the Transformer Encoder is then fed into the head of MLP for detection or classification and prediction process. The Transformer Encoder, as proposed by previous research [14], cannot be separated from the nature of the Transformer, namely the attention mechanism. The attention mechanism has an architecture, as shown in Fig. 3.

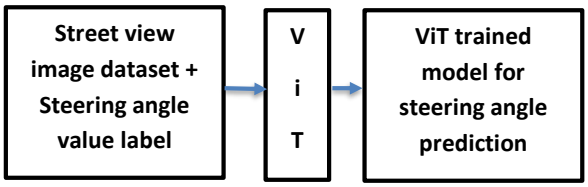


Fig. 4. ViT training process.

The attention mechanism performs vector mapping of each image patch by weighting queries (Q), keys (K), and values (V). The output of the attention mechanism is the scaled-dot-product. It can be calculated using Eq. (5) [19].

$$\text{Attention}\,(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \quad (5)$$

Next, the use of MHA which is useful for improving the performance of attention mechanisms is carried out by combining attention mechanisms in parallel. MHA can be calculated using Eq. (6). It shows W as the weight matrices.

$$\begin{aligned}\text{MultiHead}(Q,K,V) &= \text{Concat}\\ &(\text{head}_1,\text{head}_2,\ldots,\text{head}_h)W^O,\\ \text{head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V). \quad (6)\end{aligned}$$

ViT must be trained using a street image dataset equipped with the actual steering angle label value to obtain a steering angle prediction model based on street image. Then, it can determine the steering angle in each image captured by the camera. Details of the ViT training process to predict steering angle can be seen in Fig. 4.

The image used for the training process must be preprocessed to remove unnecessary parts, such as trees, buildings, and the sky, so that the prediction process focuses more on the road and the objects in it. In addition, resizing is also done to reduce the pixel size of the image dataset without losing important information in it. This resizing is useful for saving storage space and speeding up the training process. In the preprocessing step, image augmentation, such as darkening and blurring, is also carried out to add to the dataset combination. Hence, the model can predict the steering angle at night, in bad weather, or on dusty roads well. By adding these combinations, the model can generalize better [25]. The complete preprocessing steps can be seen in Fig. 5.

After the ViT model is trained, the ViT model can predict the steering angle. It is according to the order of the incoming street image sequence captured via a front-facing camera on an autonomous car, as shown in Fig. 6.

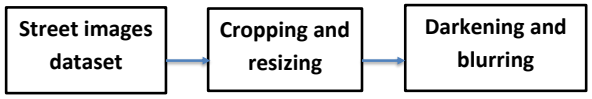


Fig. 5. Preprocessing steps.

Fig. 6. ViT model for steering angle prediction.

III. Results and Discussion

The results of the seven experiments are presented. The first five experiments predict steering angle using ViT and its modifications to find the most optimal model. The most optimal model is compared with two existing models. It consists of the CNN model of end-to-end steering angle prediction developed by NVIDIA [5] and the end-to-end CNN-LSTM combination model developed by previous research [12] through the sixth and seventh experiments.

The ViT model used is the 12-layer ViT-Base model developed [19]. ViT-Base has 12 layers with a total hidden size of 768, MLP size of 3.072, 12 heads, and 86M parameters. The reason for using this model is because it has the fewest layers, with 12 layers. Moreover, the modifications use 9 Transformer Encoder layers until 5 Transformer Encoder layers with 4 heads. The expected optimal model has fewer layers and can reduce the resulting latency. It is very suitable for the steering angle prediction process, considering that the amount of image data captured by the autonomous car camera will change very quickly sequentially.

A public dataset (https://github.com/SullyChen/driving-datasets) is used for the training process. The dataset is taken from streets around Rancho Palos Verdes and San Pedro, California. The number of images is 45,560, which have been labeled with the steering angle value for each image, as shown in Fig. 7. Each image is 455×256 pixels.

Next, Fig. 8 shows the results of preprocessing the dataset. The original image from the data set is cropped and reduced in pixel size so that only the road image remains. Then, the augmentation process is carried out through darkening and blurring.

Using an Intel Core i7-3632QM 2.2GHz CPU, 8GB RAM, and NVIDIA GeForce GT 620M GPU, the training process lasts seven hours until a ViT model is obtained to predict steering angle. The algorithm used for ViT can be seen in Algorithm 1. The dataset used is divided into several batch sizes. Then, each image in

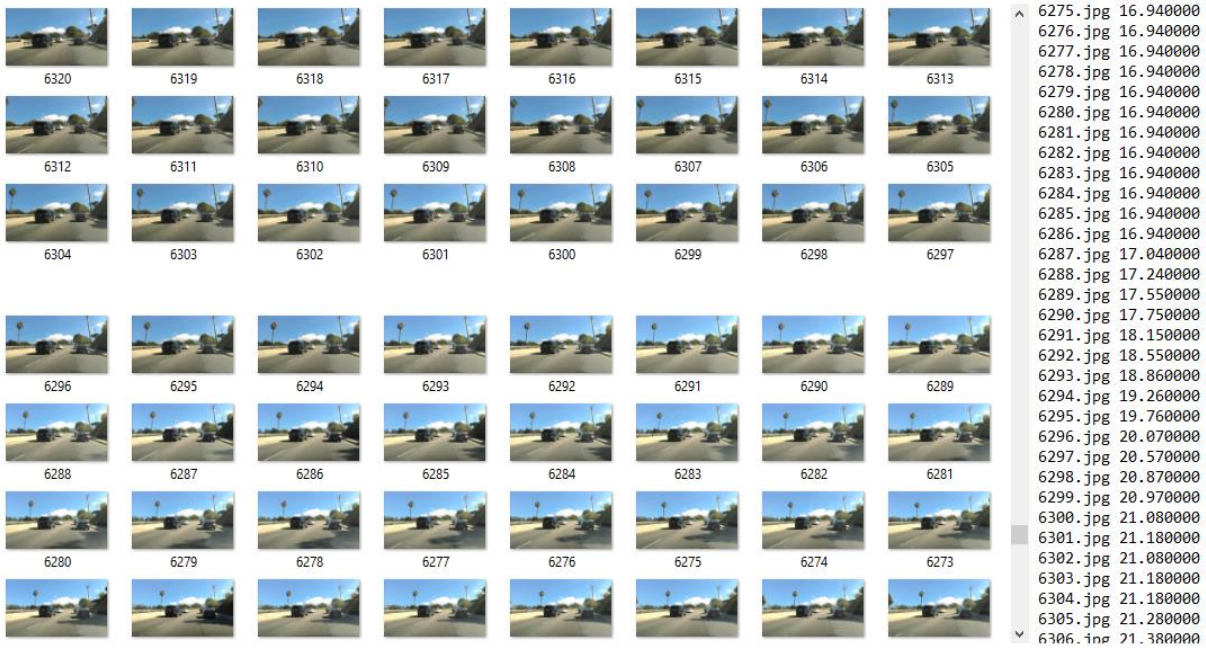


Fig. 7. Image datasets and label examples.

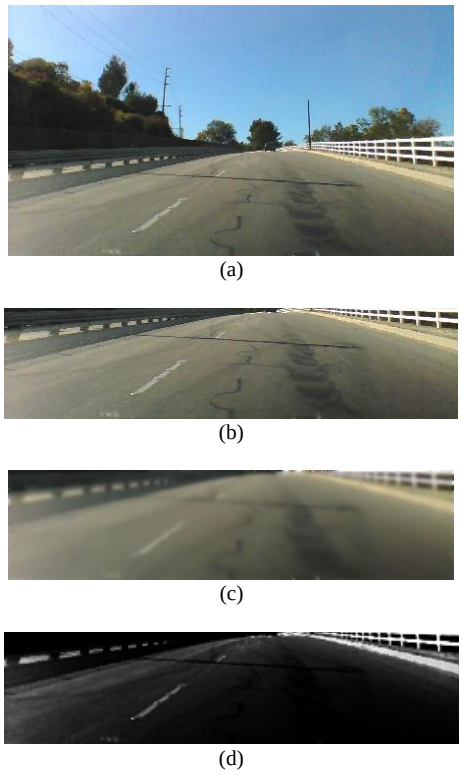


Fig. 8. Dataset preprocessing results: (a) original, (b) cropped, (c) blur, and (d) dark.

Algorithm 1 Vision Transformer

Vision Transformer (ViT)

Input: Street images with labels $\{x_i y_i\}_{i=1}^{n}$

Output: Predicted steering angle labels from the Street images test

1. Set epoch to 300, set batch size to 25, learning rate to 0.001, Loss function Sparse Categorical Cross Entropy, activation function Adam
2. Set image size to 32, patch size to 6, number of patch = *(image size : patch size)*2*
3. Augmentation Process
4. For epoch = 1: number of epochs
   - 4.1. For batch = 1: number of batches of training datasets
     - Generate another batch of training datasets.
     - Train the ViT model.
     - Backpropagate the loss.
     - Update ViT parameter.
   - 4.2. Update steering angle model prediction
5. Steering angle label prediction

the dataset is reduced in size to speed up the training process without losing important features in it.

Figure 9 shows the results of the image patch that has been done. The image size is changed to 32×32, and the patch size used is 6×6. It results in 25 patches per image and 108 elements per image. Using a smaller patch size increases the sequence length, which affects latency [22].

Using the Adam activation function and Sparse

Fig. 9. Patch image results.

TABLE I
TRAINING AND VALIDATION RESULTS.

	Model	Training		Validation	
		Accuracy	Loss	Accuracy	Loss
1	ViT-Base modified to 4 heads and 9 Transformer Encoder layers	0.778	0.506	0.787	0.461
2	ViT-Base modified to 4 heads and 8 Transformer Encoder layers	0.812	0.403	0.797	0.399
3	ViT-Base modified to 4 heads and 7 Transformer Encoder layers	0.813	0.405	0.810	0.413
4	ViT-Base modified to 4 heads and 6 Transformer Encoder layers	0.813	0.416	0.799	0.548
5	ViT-Base modified to 4 heads and 5 Transformer Encoder layers	0.803	0.395	0.792	0.535

TABLE II
FPS RESULTS.

	Model	FPS
1	9 Transformer Encoder layers	23
2	8 Transformer Encoder layers	24
3	7 Transformer Encoder layers	24
4	6 Transformer Encoder layers	25
5	5 Transformer Encoder layers	25

Categorical Cross Entropy loss function, 80% of the dataset is used for training, 20% is for validation, and the number of epochs used is 300. The training and validation results can be seen in Table I and Fig. A1 in Appendix. The training results are highly convergent for the modified ViT-Base with 8 Transformer Encoder layers and the modified ViT-base model into 7 Transformer Encoder layers. There is no underfitting [26] or overfitting [27]. It shows that the model can generalize well [28]. In addition, there is a slight overfitting for the modified ViT-Base model into 9, 6, and 5 Transformer Encoder layers. Of the five models, modified ViT-Base to 7 Transformer Encoder layers is the most optimal model with the highest training and validation accuracy value. The following experiment will use a modified ViT-Base in 7 Transformer Encoder layers.

The examples of visualization results from the attention map of each image can be seen in Fig. A2 in Appendix. The image on the left shows the original image, while the image on the right shows the attention map. The classification results focus on street views on the road, such as road markings, traffic lights, road fences, and cars. It can happen due to the augmentation process during the training process.

Next, using a simple Python program, a simulator is created to measure the performance of the proposed model. This simulator does not use live images taken through the camera but uses dataset images that have been used for the training process. The simulator displays dataset images sequentially to be displayed like a video. It also displays the actual steering angle values and predicts steering angle values complete with steering wheel images to visualize steering angle predictions according to the sequence images displayed. The simulator display can be seen in Fig. A3 in Appendix.

The resulting Frames per Second (FPS) are also compared for the five ViT models to see the fastest processing time. The results are in Table II. The resulting video output can reach more than 15 FPS for all models. The modified ViT-Base into 6 and 5 Transformer Encoder layers have the highest FPS due to the least number of layers. Thus, the prediction results are real-time, so there are no problems with processing time [29].

Next, the steering angle prediction results and its comparison with the actual value can be seen in Fig. 10. From the results of the graph comparison, it can be seen that the predicted results of the steering angle are close to the actual results based on the dataset labels in each image.

Using the Mean Squared Error (MSE) calculation according to Eq. (7), the MSE value is 2.991. It shows

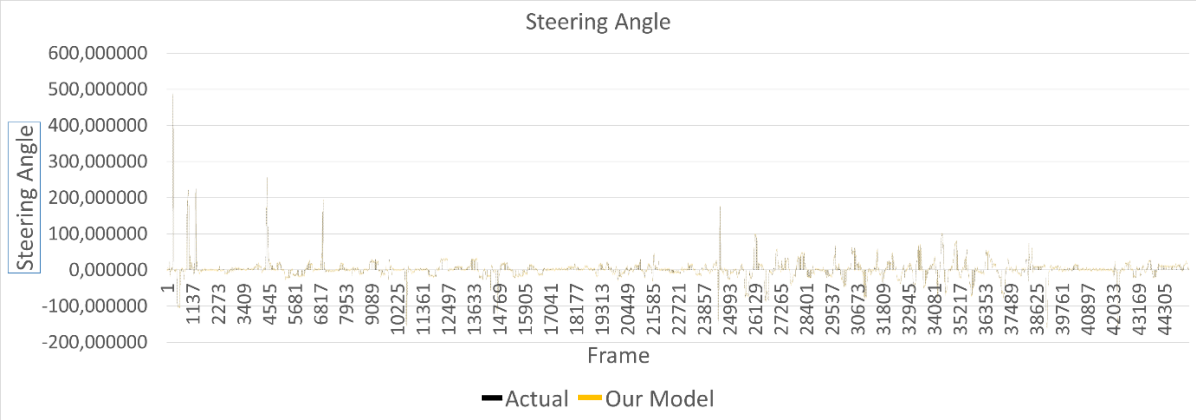


Fig. 10. Steering angle prediction result from the proposed model vs. actual steering angle.

n as the number of frames, Y as the actual value of the steering angle, and $\hat{Y}$ as the prediction value of the steering angle. This result is compared with existing models to see the level of accuracy of this proposed model.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2. \tag{7}$$

A comparison of the MSE values is carried out using two previously developed models, namely the model developed by NVIDIA [5] and the model developed by previous research [12], to find out the performance of the proposed model more objectively. The NVIDIA model uses a CNN model with nine layers consisting of one LN, five convolutional layers, and three fully connected layers. Meanwhile, the model by previous research [12] has a combination of CNN with LSTM. CNN extracts spatial features from image input sequences, and LSTM captures temporal information from the image input. The network consists of five convolutional layers, an LSTM layer, and three fully connected layers.

The previous research [12] also compares the results with a model developed by NVIDIA using real-world datasets from Udacity. Although the research also makes a comparison between the model proposed by Jiang et al. [12] and developed by NVIDIA indirectly, the datasets used are different and may affect accuracy [22, 30]. The results of comparing the predicted steering angles of the three models with the actual values can be seen in Table III and Fig. 11.

From Table III , the proposed model has the lowest MSE value. It means the model has higher accuracy than the model proposed by NVIDIA and Jiang et al. [12] using the same dataset. The FPS generated are all over 15 FPS, so there is no problem with processing time. It can be used in real-time for autonomous cars [29]. The sequence-to-sequence model from ViT can be used to process time series image data to predict the steering angle of an autonomous model in real-time. It has higher accuracy than the CNN model developed by NVIDIA and the combination of CNN and LSTM developed by Jiang et al. [12].

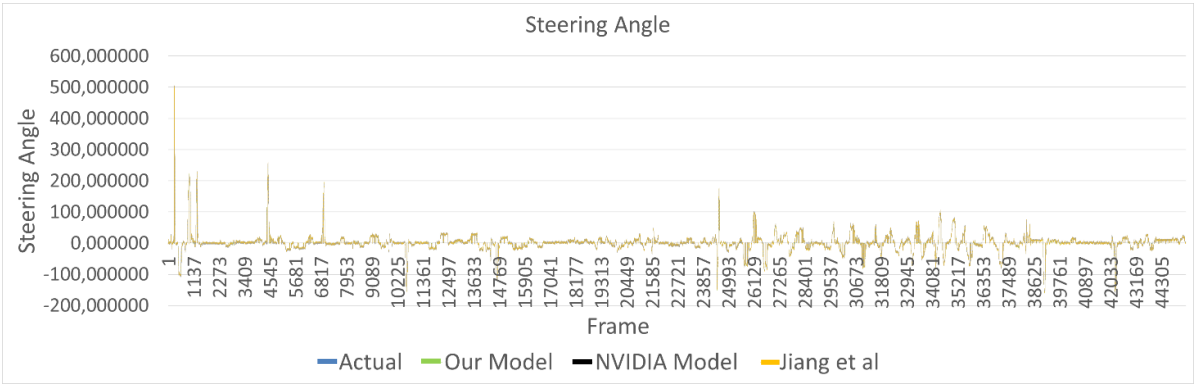
TABLE III
COMPARISON RESULTS BETWEEN THE PROPOSED AND PREVIOUS MODELS.

No	Model	MSE	FPS
1	The proposed model	2.991	24
2	Jiang et al. [12]	4.065	24
3	NVIDIA	5.358	25

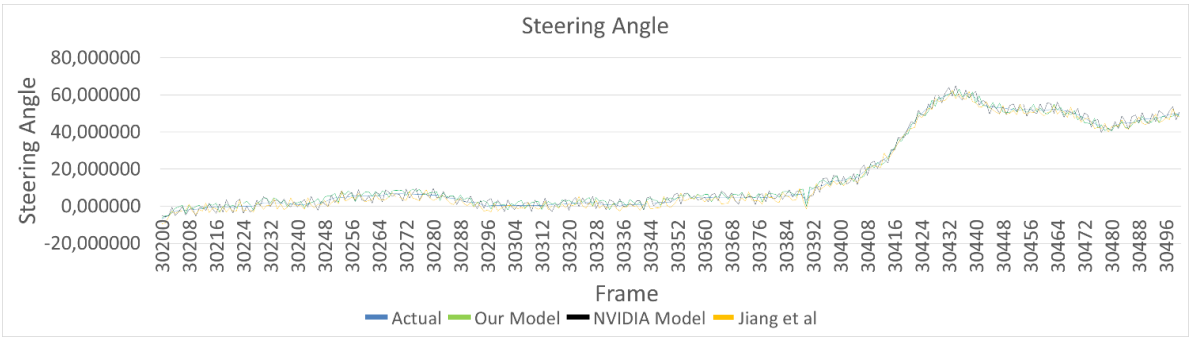
IV. CONCLUSION

The proposed model can predict steering angle well with an MSE value of 2.991 through a modified ViT-Base model into 7 Transformer Encoder layers. Using the same dataset, the proposed model gets better MSE results compared to the model proposed by NVIDIA and previous research. It is well received in terms of processing time as it can reach 24 FPS for the research model and model in previous research. Meanwhile, the model developed by NVIDIA reaches 25 FPS. From the results of this experiment, the ViT model can be used to replace the existing model, namely the CNN model and the combination model between CNN and LSTM, in predicting the steering angle of an autonomous car.

As a research limitation, this model is developed in a simulator environment. Although this simulator uses real road videos, it needs further adjustments, such as model response speed in predicting steering angles based on road conditions when applied to real



(a)



(b)

Fig. 11. Comparison of the predicted steering angle results from the three models with the actual value: (a) all frame and (b) frame number 30,200–30,500 in detail.

autonomous cars. It is possible to add speed labels to the dataset used. The model can predict not only the steering angle but also the expected speed using dual MLP outputs.

Further research can also be carried out using datasets for different locations with more complex object complexity on the street. For example, it can use a larger number of vehicles and pedestrians. In addition, further experiments can also be carried out during rainy or dusty road conditions to see the resulting prediction in predicting the steering angle.

REFERENCES

[1] P. Penmetsa, E. K. Adanu, D. Wood, T. Wang, and S. L. Jones, "Perceptions and expectations of autonomous vehicles–A snapshot of vulnerable road user opinion," *Technological Forecasting and Social Change*, vol. 143, pp. 9–13, 2019.

[2] T. Sawabe, M. Kanbara, and N. Hagita, "Comfort intelligence for autonomous vehicles," in *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. Munich, Germany: IEEE, Oct. 16–20, 2018, pp. 350–353.

[3] U. M. Gidado, H. Chiroma, N. Aljojo, S. Abubakar, S. I. Popoola, and M. A. Al-Garadi, "A survey on deep learning for steering angle prediction in autonomous vehicles," *IEEE Access*, vol. 8, pp. 163 797–163 817, 2020.

[4] S. Kuutti, R. Bowden, Y. Jin, P. Barber, and S. Fallah, "A survey of deep learning applications to autonomous vehicle control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 712–733, 2020.

[5] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," 2016. [Online]. Available: https://arxiv.org/abs/1604.07316

[6] H. Zhang, J. Bosch, and H. H. Olsson, "End-to-end federated learning for autonomous driving vehicles," in *2021 International Joint Conference on Neural Networks (IJCNN)*. Shenzhen, China:

IEEE, July 18–22, 2021, pp. 1–8.

[7] S. Lade, P. Shrivastav, S. Waghmare, S. Hon, S. Waghmode, and S. Teli, "Simulation of self driving car using deep learning," in *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*. Pune, India: IEEE, March 5–7, 2021, pp. 175–180.

[8] Y. Zhao and Y. Chen, "End-to-end autonomous driving based on the convolution neural network model," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. Lanzhou, China: IEEE, Nov. 18–21, 2019, pp. 419–423.

[9] C. Yang, W. Jiang, and Z. Guo, "Time series data classification based on dual path CNN-RNN cascade network," *IEEE Access*, vol. 7, pp. 155 304–155 312, 2019.

[10] H. Zhang, H. Lu, and A. Nayak, "Periodic time series data analysis by deep learning methodology," *IEEE Access*, vol. 8, pp. 223 078–223 088, 2020.

[11] M.-j. Lee and Y.-g. Ha, "Autonomous driving control using end-to-end deep learning," in *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*. Busan, Korea (South): IEEE, Feb. 19–22 2020, pp. 470–473.

[12] H. Jiang, L. Chang, Q. Li, and D. Chen, "Deep transfer learning enable end-to-end steering angles prediction for self-driving car," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. Las Vegas, NV, USA: IEEE, Oct. 19–Nov. 13, 2020, pp. 405–412.

[13] Z. Liu, K. Wang, J. Yu, and J. He, "End-to-end control of autonomous vehicles based on deep learning with visual attention," in *2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI)*. Hangzhou, China: IEEE, Dec. 18–20, 2020, pp. 584–589.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017, pp. 1–11.

[15] T. J. Sefara, S. G. Zwane, N. Gama, H. Sibisi, P. N. Senoamadi, and V. Marivate, "Transformer-based machine translation for low-resourced languages embedded with language identification," in *2021 Conference on Information Communications Technology and Society (ICTAS)*. Durban, South Africa: IEEE, March 10–11, 2021, pp. 127–132.

[16] A. Tjandra, S. Sakti, and S. Nakamura, "Speech-to-speech translation between untranscribed unknown languages," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. Singapore: IEEE, Dec. 14–18, 2019, pp. 593–600.

[17] C. Tho, Y. Heryadi, I. H. Kartowisastro, and W. Budiharto, "A comparison of lexicon-based and transformer-based sentiment analysis on code-mixed of low-resource languages," in *2021 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI)*, vol. 1. Jakarta, Indonesia: IEEE, Oct. 28, 2021, pp. 81–85.

[18] K. Pipalia, R. Bhadja, and M. Shukla, "Comparative analysis of different transformer based architectures used in sentiment analysis," in *2020 9th International Conference System Modeling and Advancement in Research Trends (SMART)*. Moradabad, India: IEEE, Dec. 4–5, 2020, pp. 411–415.

[19] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020. [Online]. Available: http://arxiv.org/abs/2010.11929

[20] T. Panboonyuen, S. Thongbai, W. Wongweeranimit, P. Santitamnont, K. Suphan, and C. Charoenphon, "Object detection of road assets using transformer-based YOLOX with feature pyramid decoder on Thai highway panorama," *Information*, vol. 13, no. 1, pp. 1–12, 2021.

[21] Z. Zhao, X. Wu, and H. Liu, "Vision transformer for quality identification of sesame oil with stereoscopic fluorescence spectrum image," *LWT*, vol. 158, pp. 1–9, 2022.

[22] Y. Bazi, L. Bashmal, M. M. A. Rahhal, R. A. Dayil, and N. A. Ajlan, "Vision transformers for remote sensing image classification," *Remote Sensing*, vol. 13, no. 3, pp. 1–19, 2021.

[23] R. Atienza, "Vision transformer for fast and efficient scene text recognition," in *International Conference on Document Analysis and Recognition*. Lausanne, Switzerland: Springer, Sept. 5–10, 2021, pp. 319–334.

[24] M. Zeineldeen, A. Zeyer, R. Schlüter, and H. Ney, "Layer-normalized LSTM for hybrid-HMM and end-to-end ASR," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Barcelona, Spain: IEEE, May 4–8, 2020, pp. 7679–7683.

[25] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in im-

age classification problem," in *2018 International Interdisciplinary PhD Workshop (IIPhDW)*. Poland: IEEE, May 9–12, 2018, pp. 117–122.

[26] S. Narayan and G. Tagliarini, "An analysis of underfitting in MLP networks," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2. Montreal, QC, Canada: IEEE, July 31–Aug. 4, 2005, pp. 984–988.

[27] H. Li, J. Li, X. Guan, B. Liang, Y. Lai, and X. Luo, "Research on overfitting of deep learning," in *2019 15th International Conference on Computational Intelligence and Security (CIS)*. Macao, China: IEEE, Dec. 13–16, 2019, pp. 78–81.

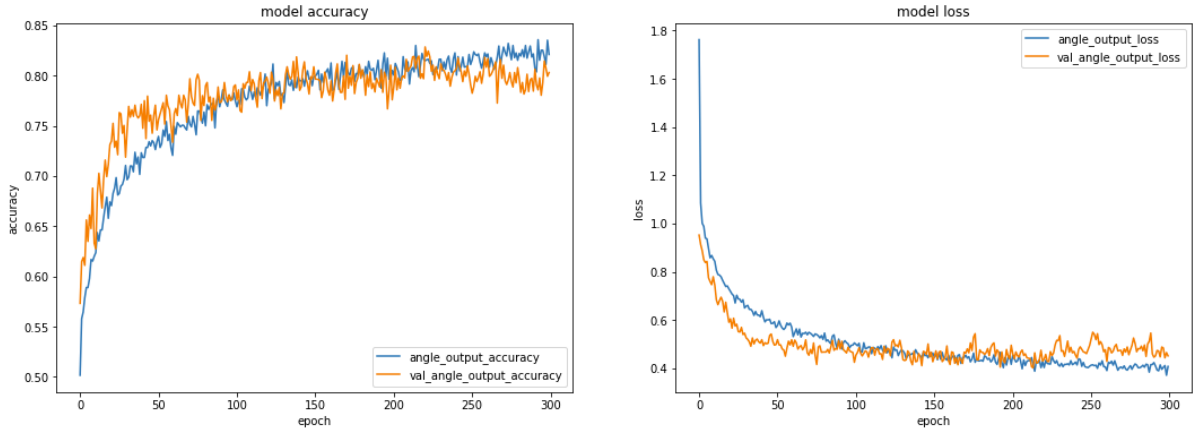
[28] J. Kolluri, V. K. Kotte, M. S. B. Phridviraj, and S. Razia, "Reducing overfitting problem in machine learning using novel L1/4 regularization method," in *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*. Tirunelveli, India: IEEE, June 15–17, 2020, pp. 934–938.

[29] J. Y. C. Chen and J. E. Thropp, "Review of low frame rate effects on human performance," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 37, no. 6, pp. 1063–1076, 2007.

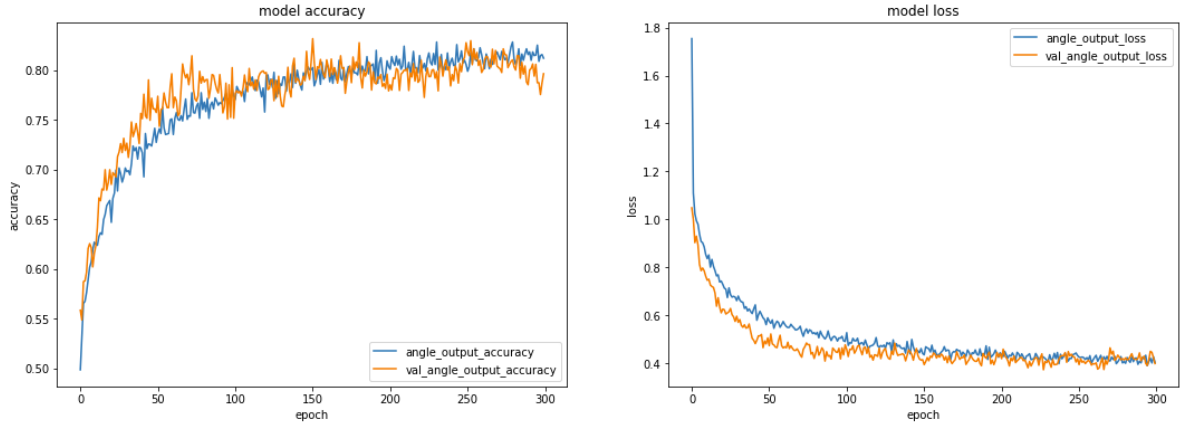
[30] K. Gauen, R. Dailey, J. Laiman, Y. Zi, N. Asokan, Y. H. Lu, G. K. Thiruvathukal, M. L. Shyu, and S. C. Chen, "Comparison of visual datasets for machine learning," in *2017 IEEE International Conference on Information Reuse and Integration (IRI)*. San Diego, CA, USA: IEEE, Aug. 4–6, 2017, pp. 346–355.

APPENDIX

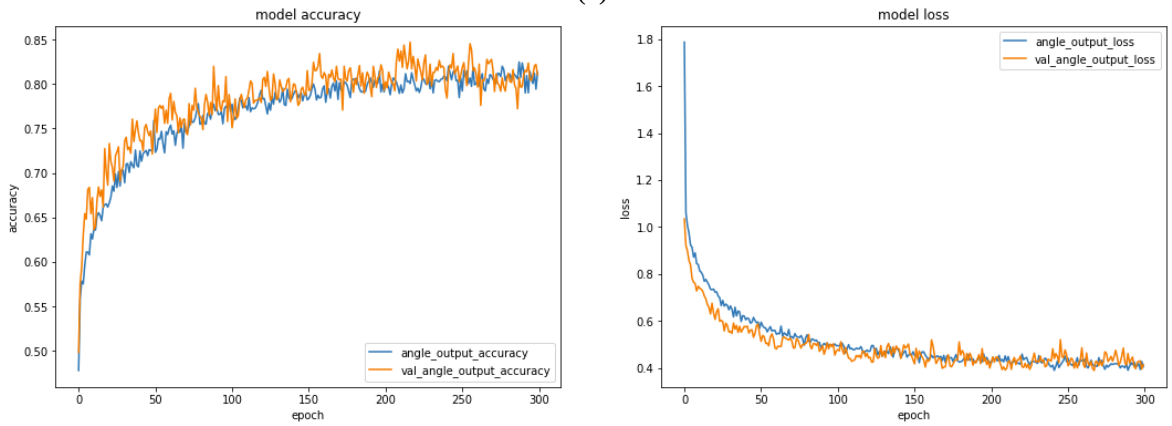
The Appendice can be seen in the next page.


model accuracy
accuracy
epoch
angle_output_accuracy
val_angle_output_accuracy
model loss
loss
epoch
angle_output_loss
val_angle_output_loss

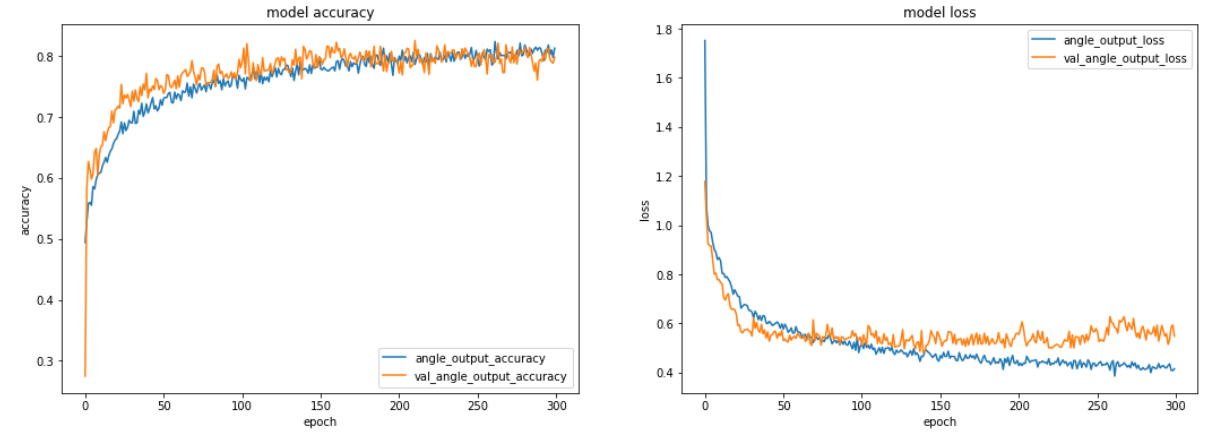
(a)


model accuracy
accuracy
epoch
angle_output_accuracy
val_angle_output_accuracy
model loss
loss
epoch
angle_output_loss
val_angle_output_loss

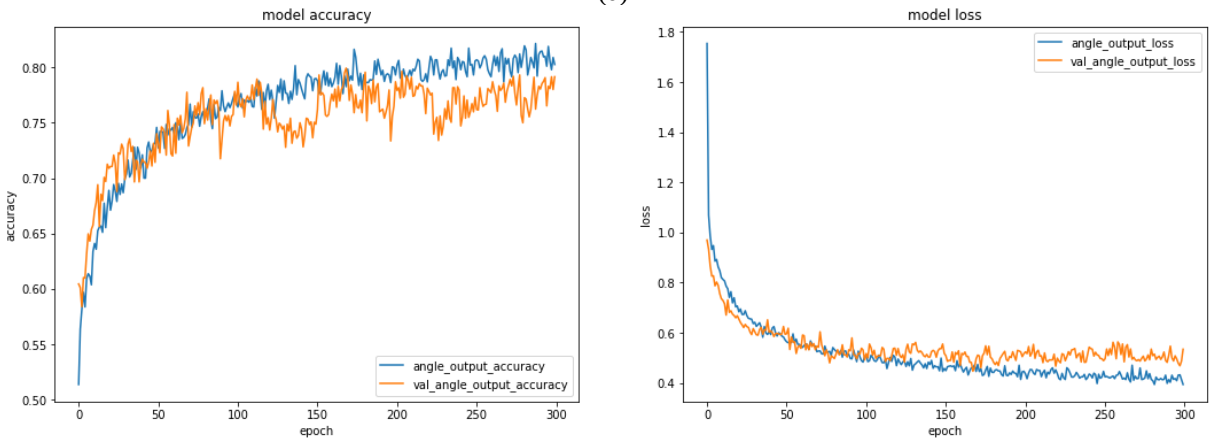
(b)


model accuracy
accuracy
epoch
angle_output_accuracy
val_angle_output_accuracy
model loss
loss
epoch
angle_output_loss
val_angle_output_loss

(c)



(d)



(e)

Fig. A1. Training and validation results: (a) 9 layers, (b) 8 layers, (c) 7 layers, (d) 6 layers, and (e) 5 layers.

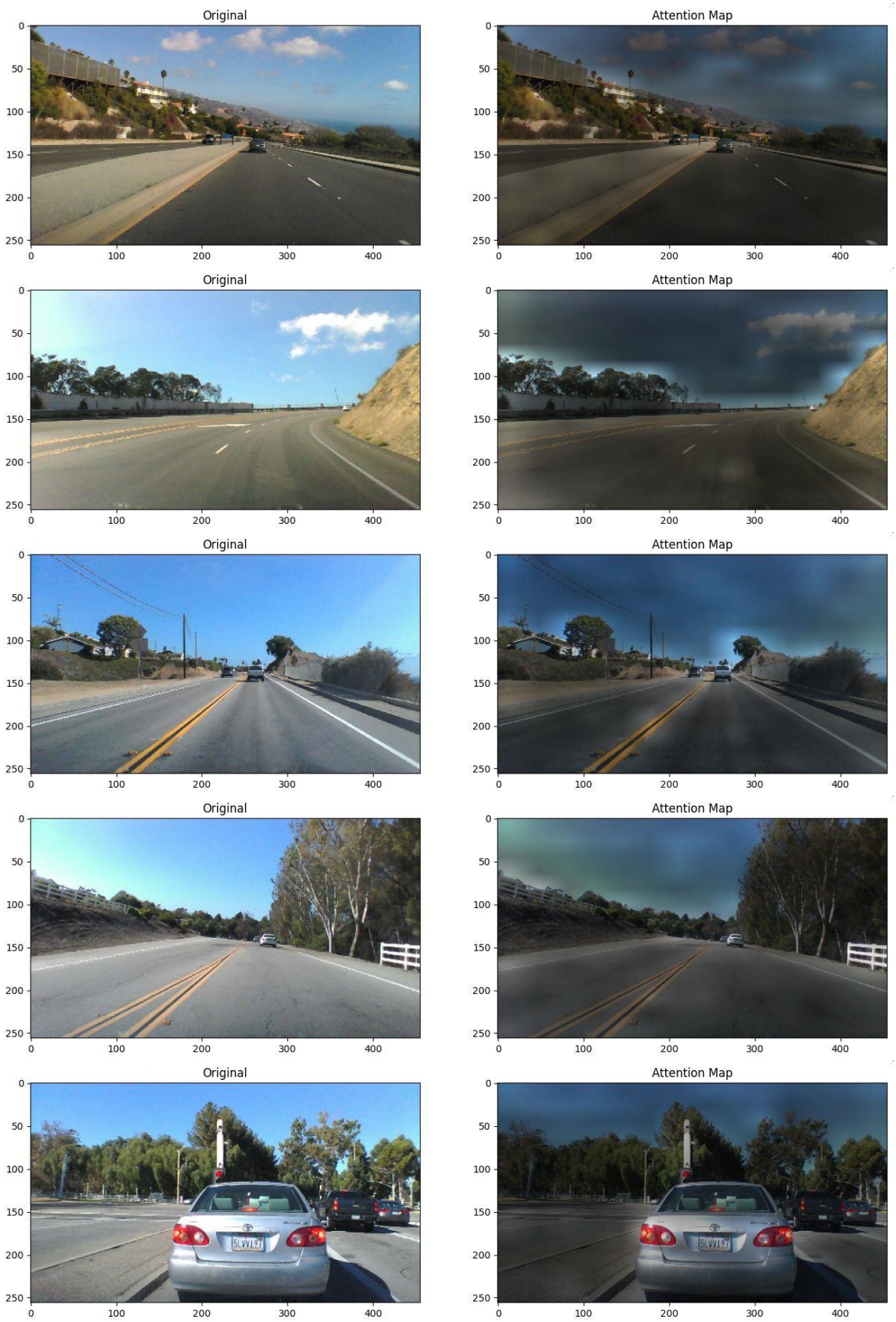


Fig. A2. Attention map.

(a)

(b)

(c)

Fig. A3. Autonomous car simulator: (a) straight ahead; (b) turn left; (c) turn right.