ICIC International ©2023 ISSN 1881-803X
pp. **349–357**

STEERING ANGLE PREDICTION FOR AUTONOMOUS CAR USING VISION TRANSFORMER

Ilvico Sonata[1,*], Yaya Heryadi[1], Antoni Wibowo[1]
and Widodo Budiharto[2]

[1]Computer Science Department, BINUS Graduate Program – Doctor of Computer Science
[2]Computer Science Department, School of Computer Science
Bina Nusantara University
Jl. K. H. Syahdan No. 9, Kemanggisan, Palmerah, Jakarta 11480, Indonesia
{ yayaheryadi; anwibowo; wbudiharto }@binus.edu
*Corresponding author: ilvico@binus.ac.id

Received March 2022; accepted June 2022

Abstract. *The development of autonomous cars today cannot be separated from the use of deep learning models. The deep learning model that is often used is the convolutional neural network (CNN) model to detect objects on the street such as cars, traffic signs, road markings and pedestrians, as well as to predict the steering angle of autonomous car. On the other hand, the development of the Transformer model as object detection and classification has also shown a rapid increase since the Transformer model was first introduced in 2017. Several previous studies have shown that the use of Transformers results in better accuracy compared to CNN. In this paper, we will discuss the use of the Transformer model to detect objects on the street and to predict the steering angle of an autonomous car. The experimental results using the simulator show that the Vision Transformer (ViT) can be used properly to predict the steering angle of an autonomous car based on previously detected objects with a mean squared error (MSE) value of 0.778. This result is better when compared to the model developed by NVIDIA and the pre-trained model VGG16.*

Keywords: Autonomous car, CNN, Steering angle prediction, Vision Transformer

1. Introduction.

The increasing development of autonomous cars today cannot be separated from the role of deep learning. Deep learning is used to detect objects on the road that are in front of autonomous cars such as vehicle, pedestrians, traffic signs, and road markings [1-3]. In addition, deep learning is also used to predict steering angle [4,5]. The deep learning model is expected to improve the safety and comfort of autonomous cars [6].

Several previous studies have used CNN as a deep learning model to detect objects on the street and to predict steering angles. For instance, Kim et al. [7], Tarmizi and Aziz [8], and Xu et al. [9] used CNN in their research to detect cars that will be used in autonomous cars. Several previous studies using CNN to detect pedestrians have been carried out by Pranav and Manikandan [10], Mounsey et al. [11], and Junaid et al. [12]. Detection of traffic signs for autonomous cars has also been carried out by many studies using CNN, including by Zhou et al. [13], Vennelakanti et al. [14], and Ng et al. [15]. Road markings in the form of lane lines detection have also been carried out by many previous studies using CNN, including by Li et al. [16], Li and Li [17], and Wang et al. [18]. Several previous studies to predict steering angle have been carried out by Bojarski et al. [19], Zhang and Huang [20], and Singhal et al. [21]. In their research, they used CNN to predict steering angle.

DOI: 10.24507/icicel.17.03.349

On the other hand, Transformer since it was first introduced by Vaswani et al. [22] in 2017 which was originally used for natural language processing (NLP), has now been widely used as an object detection model. Dosovitskiy et al. [23] have introduced a Transformer model for object detection through Vision Transformer (ViT) in 2020. In their research, Dosovitskiy et al. [23] stated that ViT achieves better results compared to CNN. Several previous studies in the field of object detection using ViT have been carried out by Panboonyuen et al. [24], Zhao et al. [25], and Park et al. [26]. From the results of their research, it was found that the use of ViT is an approach using a new method with promising image generalization capabilities and is not inferior to CNN.

In this paper, we propose the use of the ViT to detect objects on the street and predict the steering angle of an autonomous car based on the detected objects. By using the ViT model in predicting the steering angle of an autonomous car, the resulting accuracy can be improved compared to the CNN model. In the next chapter, the proposed method and the results of the experiments that have been carried out will be explained.

2. **Proposed Method.** Transformers are basically a sequence-to-sequence model used for natural language processing (NLP) as first developed by Vaswani et al. [22]. Dosovitskiy et al. [23] then developed the Transformer model into a computer vision model for detecting objects, named Vision Transformer (ViT). The ViT architectural model can be seen in Figure 1.

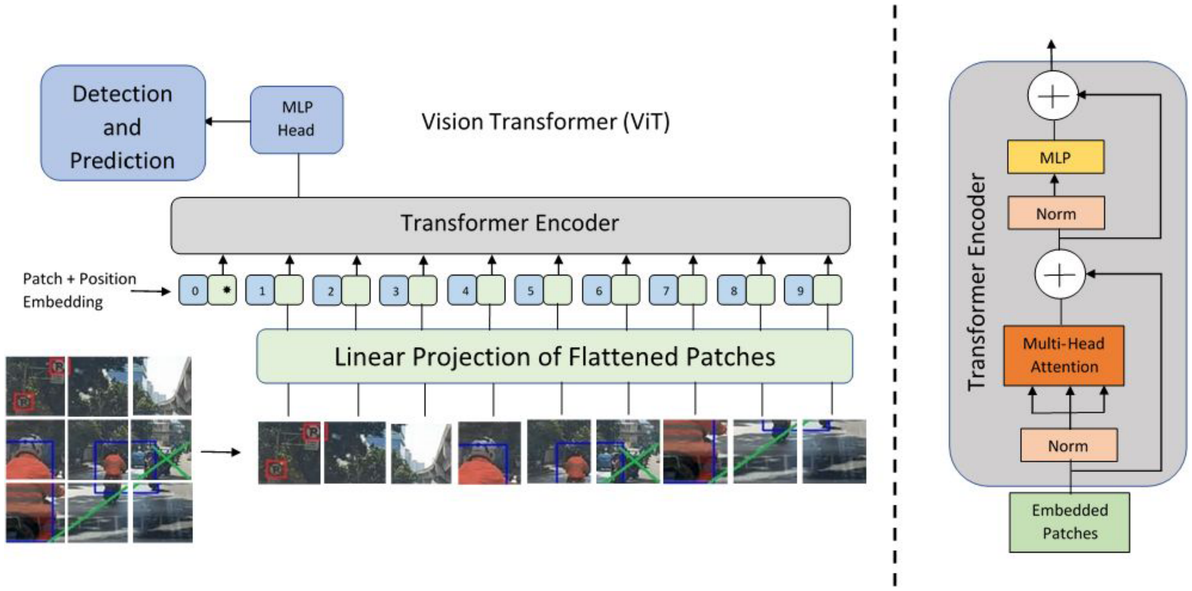


FIGURE 1. ViT architecture

The image to be processed is split into multiple patches of the same size. Each patch is then embedded into a vector set that represents each patch. To keep the original position of each patch known, its position is also embedded using position embedding so that the positions of each patch are not confused. The results of the embedding sequence are then fed into the Transformer encoder for feature extraction. The results of the feature extraction are then fed into a multi-layer perceptron for the image classification process.

The feature extraction process carried out by the Transformer encoder is inseparable from the attention mechanism that is the hallmark of the Transformer model as proposed by Vaswani et al. [22]. In general, the attention mechanism can be seen in Figure 2.

The attention mechanism performs a vector mapping of each image patch based on Query, Key, and Value and assigns a weight to each of these parameters. The output of the attention mechanism is a scaled dot-product which can be calculated using Equation (1).

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

where Q is Query, K is Key, V is Value, and d_k is the dimension of the Key.

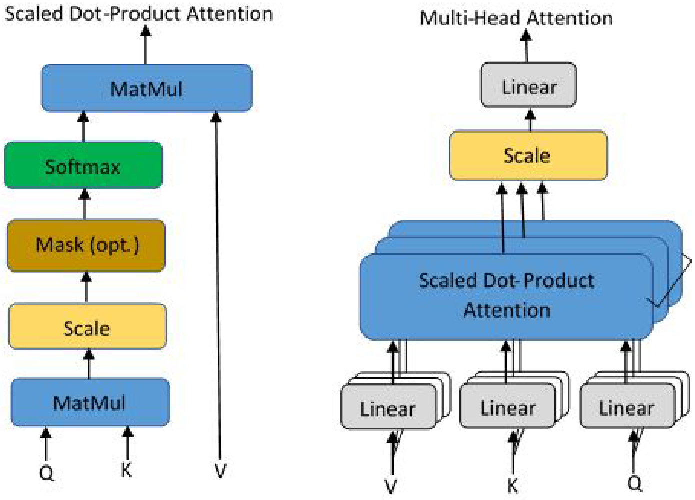


Figure 2. Attention mechanism

To improve its performance, linear projection is used to form multiple attention. This multiple attention is also called multi-head attention which can be calculated using Equations (2) and (3).

$$MultiHead(Q, K, V) = Concat\left(head_1, head_2, \ldots, head_h\right)W^O \tag{2}$$

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right), \quad W_i^Q \in \mathbb{R}^{d_{model} \times d_q}, \quad W_i^K \in \mathbb{R}^{d_{model} \times d_k}, \\ W_i^V \in \mathbb{R}^{d_{model} \times d_v}, \quad W^O \in \mathbb{R}^{hd_v \times d_{model}} \tag{3}$$

where W is the trainable weight parameter matrices, d_q is dimension of the Queries, d_v is dimension of the Value, d_{model} is the final output feature dimension, and h is the number of heads. In general, $d_q = d_k = d_v = d_{model}/h$.

Finally, the output of feed-forward multi-layer perceptron (FF-MLP) can be calculated using Equation (4).

$$\text{FF-MLP} = \sum_i \text{GELU}\left(q_i k_i^T + b_i\right) v_i + c \tag{4}$$

where b and c are bias, and the activation function used is GELU.

The proposed framework of the ViT model for predicting the steering angle of an autonomous car can be seen in Figure 3.

The first stage of the framework as shown in Figure 3 is the process of detecting objects on the street. In this paper, the objects to be detected are car, motorbike, pedestrians, traffic signs and lane lines. To be able to recognize these objects, the ViT model must be trained first using the necessary datasets, namely the dataset of cars, motorbike, pedestrians, traffic signs, and lane lines. More details about the training process can be seen in Figure 4.

Before the training process, the image from the datasets must be pre-processed first to remove if there are unnecessary parts such as trees, buildings, and the sky through the cropping process. The resizing process is used to reduce the image size without losing important information in it, thereby making processing time faster. In the pre-processing, dark and blurring processes are also carried out to add a combination of datasets during the training process so that the model can detect objects at night, rainy day, and dusty roads. Through pre-processing, the object detection process can focus on the desired object so as to improve classifier performance [27]. The pre-processing steps can be seen in Figure 5.

After the training process is complete, the ViT model can recognize the desired object through the camera mounted on the autonomous car. Figure 6 shows the object detection process.

The second stage of the framework as shown in Figure 3 is to predict the steering angle of the autonomous car. The results of detecting objects on the street that have been carried out previously become a dataset equipped with an actual steering angle value label for each image. The dataset will be used in the ViT training process to predict steering angle. The ViT training process can be seen in Figure 7.

After the training process is complete, the ViT model can predict the steering angle based on the image input from the camera as shown in Figure 8.

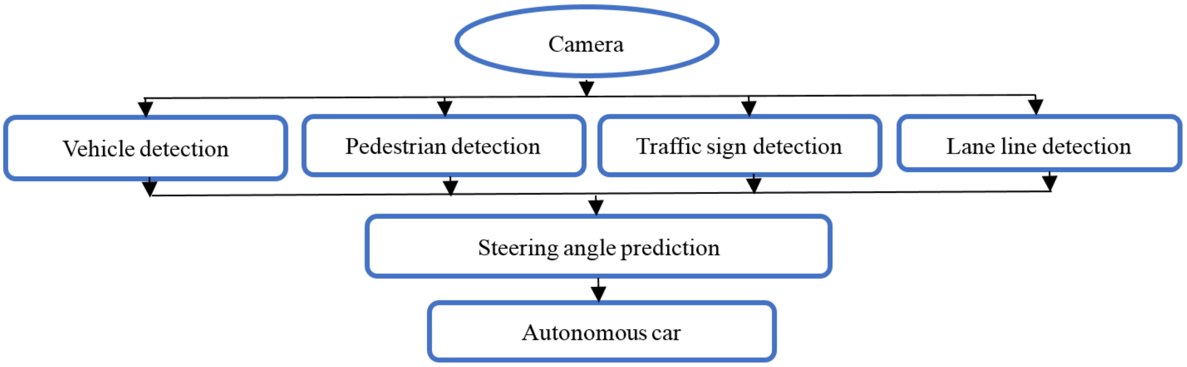


Figure 3. Proposed framework for steering angle prediction

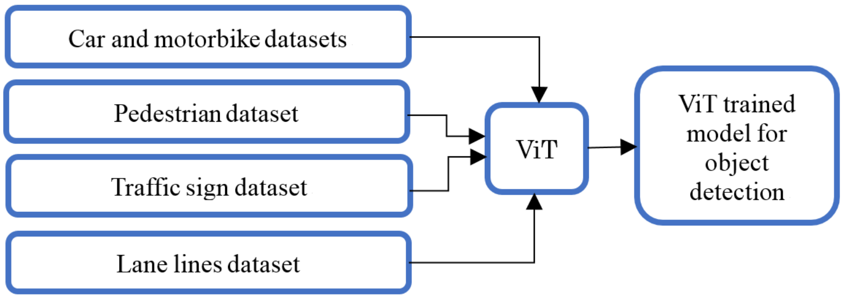

Car and motorbike datasets
Pedestrian dataset
Traffic sign dataset
Lane lines dataset
ViT
ViT trained model for object detection

Figure 4. ViT training process for object detection

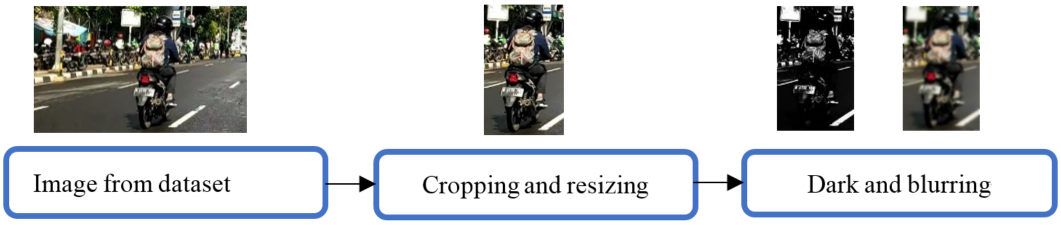


Figure 5. Image pre-processing steps

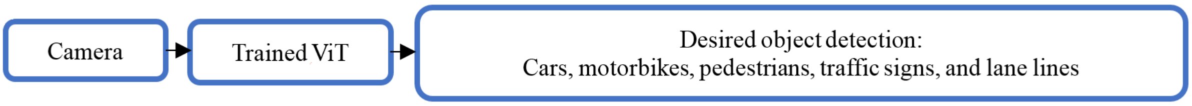

Camera
Trained ViT
Desired object detection:
Cars, motorbikes, pedestrians, traffic signs, and lane lines

Figure 6. ViT object detection process

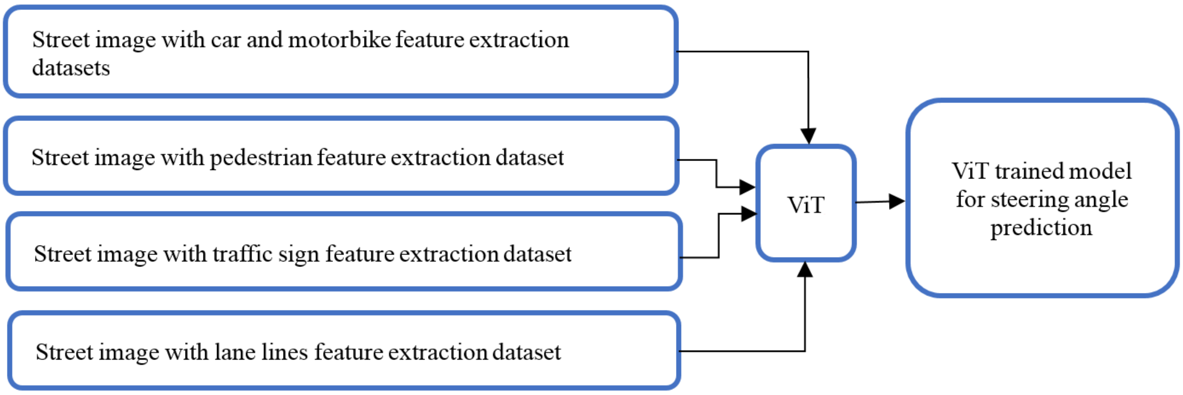

Street image with car and motorbike feature extraction datasets
Street image with pedestrian feature extraction dataset
Street image with traffic sign feature extraction dataset
Street image with lane lines feature extraction dataset
ViT
ViT trained model for steering angle prediction

Figure 7. ViT training process for steering angle prediction

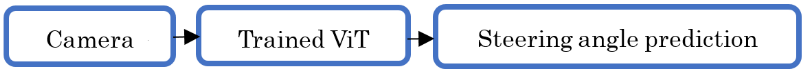


Figure 8. ViT steering angle prediction process

3. **Experimental Results.** In this chapter, two experiments were conducted: the first experiments to detect objects on the street which include cars, motorbikes, pedestrians, traffic signs and lane lines; the second experiments to predict the steering angle of an autonomous car based on the results of object detection carried out in the first experiment. Both experiments used the standard ViT model architecture ViT-Base with 9×9 input patch size with 12 layers, 768 hidden size, 3,072 MLP size, 12 heads, and 86M parameters [23].

The datasets used to train the ViT model as an object detector are an existing secondary dataset. Rearview car dataset was taken from Udacity (https://s3.amazonaws.com/udacity-sdc/Vehicle_Tracking/vehicles.zip) for a total of 834 images. Motorbike dataset was taken from http://velastin.dynu.com/videodatasets/UrbanMotorbike/mb75000.htm for a total of 7,500 images. Traffic sign dataset was taken from Kaggle (https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign) for a total of 4,272 images. Pedestrian dataset was taken from VIPeR as many as 632 images and lane line dataset was taken from the TuSimple dataset for a total of 6,408 images.

The training process used laptop with i5-4200U CPU, NVIDIA Geforce 740M GPU and 12GB memory. Using 100 epochs, the training process took about 3 hours to finish training for each classification process. Figure 9 shows the results of object detection using the previously trained ViT model. The result of object detection is a bounding box that

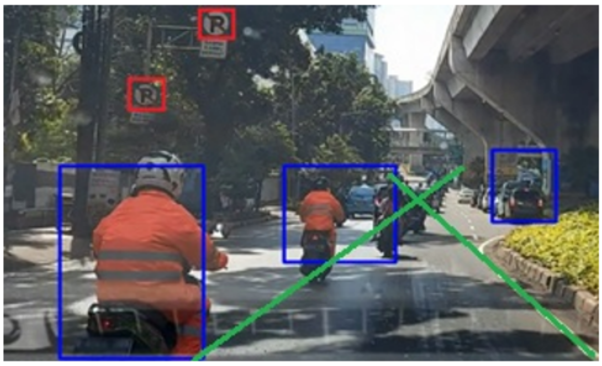
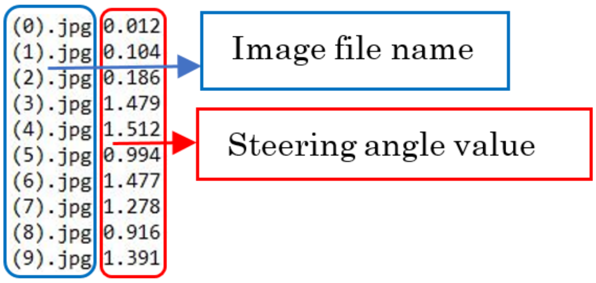


Figure 9. Object detection results

appears on the detected object. Especially for lane lines, the detection is marked with a green line.

In the second experiment, the dataset was taken from street images in Jakarta where object detection had previously been carried out through the first experiment. The dataset was collected using video to record the streets of Jakarta. The recorded video was then split into image frames with a total of 36,000 frames for a 25-minute video. Each image was labeled with a filename and steering angle value. A CSV file was generated which contains the steering angle data for each frame as shown in Figure 10.



```
(0).jpg 0.012
(1).jpg 0.104
(2).jpg 0.186
(3).jpg 1.479
(4).jpg 1.512
(5).jpg 0.994
(6).jpg 1.477
(7).jpg 1.278
(8).jpg 0.916
(9).jpg 1.391
```

Figure 10. CSV data label

The training process used Google Colaboratory and took about 7 hours to finish the training process. Using 300 epochs, the training and validation results are very convergent. The results of the validation using 20% of the image dataset show a higher accuracy value and a lower loss value compared to the training results. The value of training and validation also tends to be stable. This condition indicates the model is not underfitting or overfitting. The validation results show an accuracy of 0.865 and a loss of 0.231. With this condition, the model can be generalized to other image data from camera [28]. The training and validation results can be seen in Figure 11.

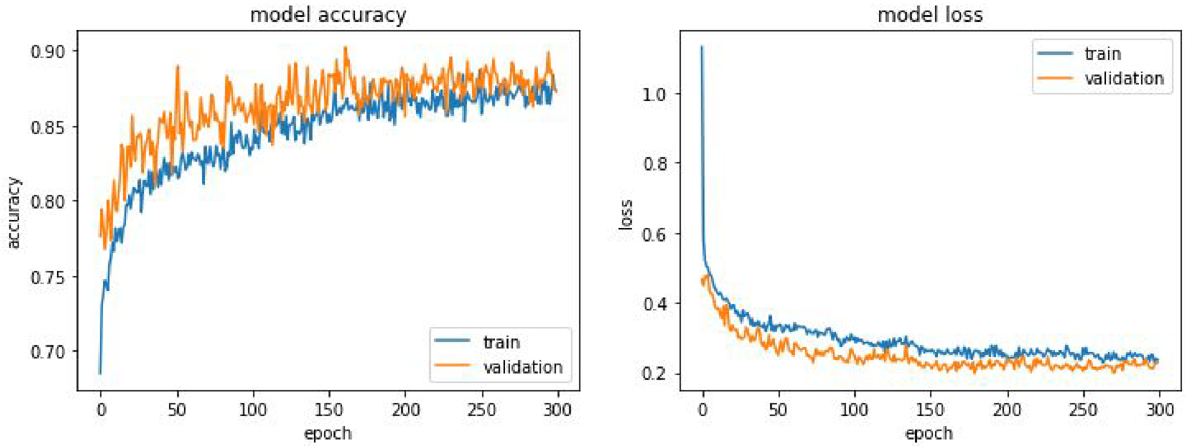
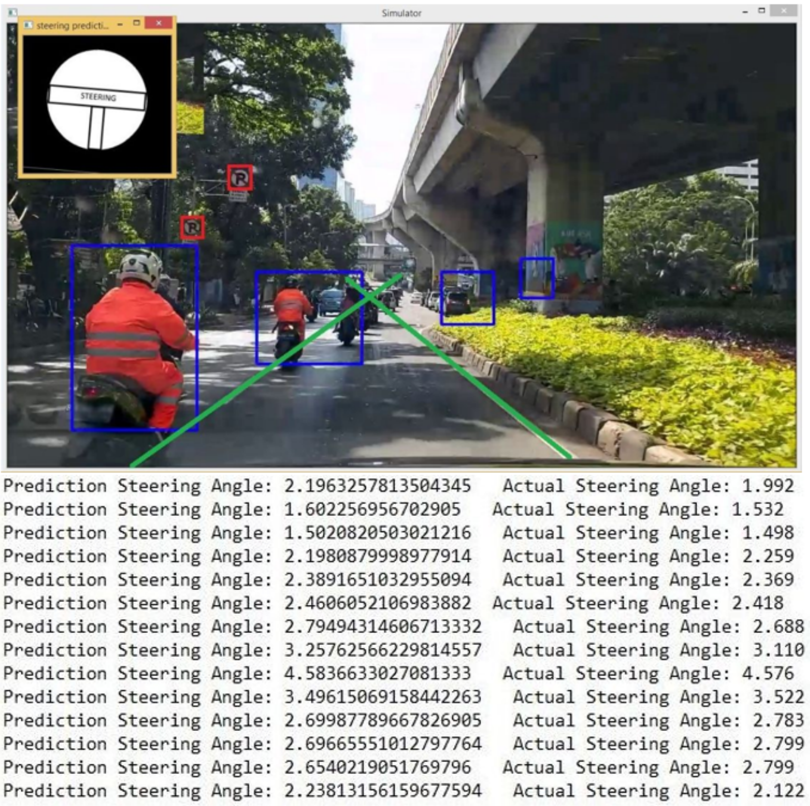


Figure 11. Training and validation results for steering angle prediction

Using a simple Python program, a simulator was created to show the position of the rotating steering angle according to the predicted results from the trained ViT model and display the predicted results and the actual value of the steering angle based on previously captured 25-minutes video recordings. The video output is 25 FPS so there is no significant screen display delay. Figure 12 shows the simulator used.

In this experiment, the predicted steering angle will be compared with the actual steering angle. Comparisons are also made with existing models. The comparison model used in this experiment was the model developed by NVIDIA [19] and the VGG16 pre-trained model using the same dataset. A graphical comparison per image frame can be seen in



```
Prediction Steering Angle: 2.1963257813504345   Actual Steering Angle: 1.992
Prediction Steering Angle: 1.602256956702905   Actual Steering Angle: 1.532
Prediction Steering Angle: 1.5020820503021216   Actual Steering Angle: 1.498
Prediction Steering Angle: 2.1980879998977914   Actual Steering Angle: 2.259
Prediction Steering Angle: 2.3891651032955094   Actual Steering Angle: 2.369
Prediction Steering Angle: 2.4606052106983882  Actual Steering Angle: 2.418
Prediction Steering Angle: 2.79494314606713332   Actual Steering Angle: 2.688
Prediction Steering Angle: 3.25762566229814557   Actual Steering Angle: 3.110
Prediction Steering Angle: 4.5836633027081333   Actual Steering Angle: 4.576
Prediction Steering Angle: 3.49615069158442263   Actual Steering Angle: 3.522
Prediction Steering Angle: 2.69987789667826905   Actual Steering Angle: 2.783
Prediction Steering Angle: 2.69665551012797764   Actual Steering Angle: 2.799
Prediction Steering Angle: 2.6540219051769796   Actual Steering Angle: 2.799
Prediction Steering Angle: 2.23813156159677594   Actual Steering Angle: 2.122
```

Figure 12. Autonomous car simulator

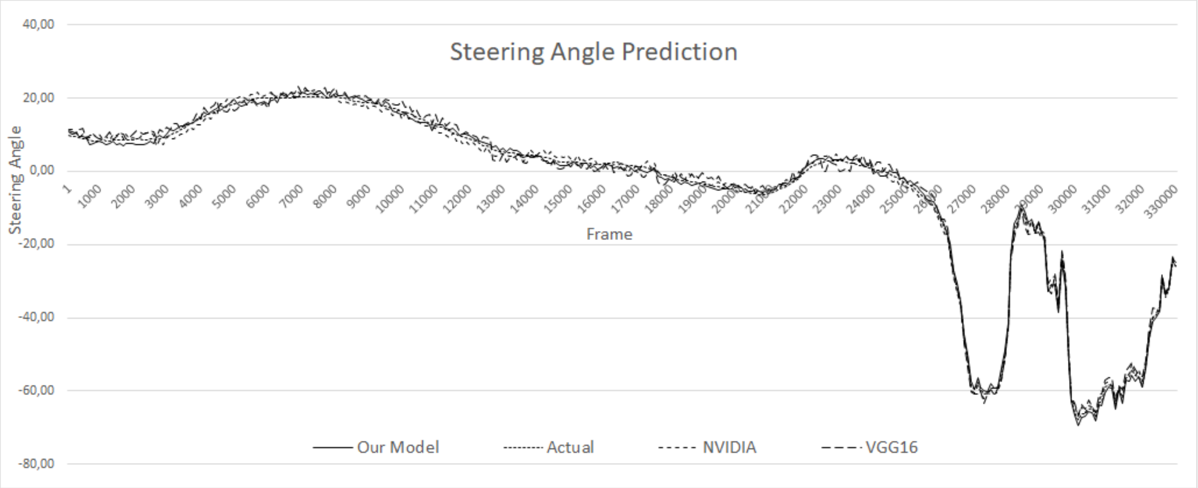


Figure 13. A graphical comparison between predicted and actual steering angle using our model, NVIDIA, and VGG16 models

Figure 13. From the graph it can be seen that the mean squared error (MSE) value obtained by our model for steering angle prediction is 0.778 compared to the NVIDIA model and the VGG16 model of 1.248 and 3.004, respectively. The comparison results can be seen in Table 1.

4. **Conclusion.** The ViT can be used properly to predict the steering angle of an autonomous car with an MSE value of 0.778. With 25 FPS video output, there is no significant delay caused by processing time. Our model yields higher accuracy compared to the existing model developed by NVIDIA [19] and the VGG16 pre-trained model. With the level of accuracy and processing time obtained in this experiment, the ViT can be used

Table 1. Comparison results

Model	MSE	FPS
Our model	0.778	25
NVIDIA	1.248	25
VGG16	3.004	24

as an alternative to the CNN model in predicting the steering angle of an autonomous car. Accuracy improvements can be made by adding datasets with more complex traffic conditions and adding detection of on-road objects such as bicycles and trucks.

Further research can be carried out by adding a speed value label to the dataset used so that the developed ViT model not only predicts the steering angle but also predicts the speed control of an autonomous car.

REFERENCES

[1] A. Uçar, Y. Demir and C. Güzeliş, Object recognition and detection with deep learning for autonomous driving applications, *Simulation*, vol.93, no.9, pp.759-769, DOI: 10.1177/0037549717709932, 2017.

[2] J. Ren, H. Gaber and S. S. Al Jabar, Applying deep learning to autonomous vehicles: A survey, *2021 4th Int. Conf. Artif. Intell. Big Data (ICAIBD2021)*, pp.247-252, DOI: 10.1109/ICAIBD51990.2021.9458968, 2021.

[3] R. Walambe, A. Marathe, K. Kotecha and G. Ghinea, Lightweight object detection ensemble framework for autonomous vehicles in challenging weather conditions, *Comput. Intell. Neurosci.*, vol.2021, DOI: 10.1155/2021/5278820, 2021.

[4] P. J. Navarro, L. Miller, F. Rosique, C. Fernández-Isla and A. Gila-Navarro, End-to-end deep neural network architectures for speed and steering wheel angle prediction in autonomous driving, *Electron.*, vol.10, no.11, pp.1-21, DOI: 10.3390/electronics10111266, 2021.

[5] 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.163797-163817, DOI: 10.1109/ACCESS.2020.3017883, 2020.

[6] K. Muhammad, A. Ullah, J. Lloret, J. Del Ser and V. H. C. De Albuquerque, Deep learning for safe autonomous driving: Current challenges and future directions, *IEEE Trans. Intell. Transp. Syst.*, vol.22, no.7, pp.4316-4336, DOI: 10.1109/TITS.2020.3032227, 2021.

[7] J. Kim, S. Hong and E. Kim, Novel on-road vehicle detection system using multi-stage convolutional neural network, *IEEE Access*, vol.9, pp.94371-94385, DOI: 10.1109/ACCESS.2021.3093698, 2021.

[8] I. A. Tarmizi and A. A. Aziz, Vehicle detection using convolutional neural network for autonomous vehicles, *Int. Conf. Intell. Adv. Syst. (ICIAS2018)*, DOI: 10.1109/ICIAS.2018.8540563, 2018.

[9] C. Xu et al., Fast vehicle and pedestrian detection using improved Mask R-CNN, *Math. Probl. Eng.*, vol.2020, DOI: 10.1155/2020/5761414, 2020.

[10] K. B. Pranav and J. Manikandan, Design and evaluation of a real-time pedestrian detection system for autonomous vehicles, *2020 Zooming Innov. Consum. Technol. Conf. (ZINC2020)*, pp.155-159, DOI: 10.1109/ZINC50678.2020.9161768, 2020.

[11] A. Mounsey, A. Khan and S. Sharma, Deep and transfer learning approaches for pedestrian identification and classification in autonomous vehicles, *Electron.*, vol.10, no.24, DOI: 10.3390/electronics10243159, 2021.

[12] M. Junaid, Z. Szalay and Á. Török, Evaluation of non-classical decision-making methods in self driving cars: Pedestrian detection testing on cluster of images with different luminance conditions, *Energies*, vol.14, no.21, pp.1-16, DOI: 10.3390/en14217172, 2021.

[13] K. Zhou, Y. Zhan and D. Fu, Learning region-based attention network for traffic sign recognition, *Sensors (Switzerland)*, vol.21, no.3, pp.1-21, DOI: 10.3390/s21030686, 2021.

[14] A. Vennelakanti, S. Shreya, R. Rajendran, D. Sarkar, D. Muddegowda and P. Hanagal, Traffic sign detection and recognition using a CNN ensemble, *2019 IEEE Int. Conf. Consum. Electron. (ICCE2019)*, pp.1-4, DOI: 10.1109/ICCE.2019.8662019, 2019.

[15] Z. B. Ng, K. M. Lim and C. P. Lee, Traffic sign recognition with convolutional neural network, *2021 9th Int. Conf. Inf. Commun. Technol. (ICoICT2021)*, pp.48-53, DOI: 10.1109/ICoICT52021.2021.9527505, 2021.

[16] J. Li, D. Zhang, Y. Ma and Q. Liu, Lane image detection based on convolution neural network multi-task learning, *Electron.*, vol.10, no.19, DOI: 10.3390/electronics10192356, 2021.

[17] H. Li and X. Li, Flexible lane detection using CNNs, *Proc. of 2021 Int. Conf. Comput. Technol. Media Converg. Des. (CTMCD2021)*, pp.235-238, DOI: 10.1109/CTMCD53128.2021.00057, 2021.

[18] W. Wang, H. Lin and J. Wang, CNN based lane detection with instance segmentation in edge-cloud computing, *J. Cloud Comput.*, vol.9, no.1, DOI: 10.1186/s13677-020-00172-z, 2020.

[19] M. Bojarski et al., End to end learning for self-driving cars, *arXiv.org*, arXiv: 1604.07316, 2016.

[20] J. Zhang and H. Huang, Steering angle prediction for autonomous cars based on deep neural network method, *2020 Aust. New Zeal. Control Conf. (ANZCC2020)*, no.11, pp.205-208, DOI: 10.1109/ANZCC50923.2020.9318380, 2020.

[21] V. Singhal, S. Gugale, R. Agarwal, P. Dhake and U. Kalshetti, Steering angle prediction in autonomous vehicles using deep learning, *Proc. of 2019 5th Int. Conf. Comput. Commun. Control Autom. (ICCUBEA2019)*, pp.1-6, DOI: 10.1109/ICCUBEA47591.2019.9128735, 2019.

[22] A. Vaswani et al., Attention is all you need, *arXiv.org*, arXiv: 1706.03762, 2017.

[23] A. Dosovitskiy et al., An image is worth 16×16 words: Transformers for image recognition at scale, *arXiv.org*, arXiv: 2010.11929, 2020.

[24] 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, *Inf.*, vol.13, no.1, DOI: 10.3390/info13010005, 2022.

[25] Z. Zhao, X. Wu and H. Liu, Vision transformer for quality identification of sesame oil with stereoscopic fluorescence spectrum image, *LWT*, vol.158, 113173, DOI: 10.1016/j.lwt.2022.113173, 2022.

[26] S. Park et al., Multi-task vision transformer using low-level chest X-ray feature corpus for COVID-19 diagnosis and severity quantification, *Med. Image Anal.*, vol.75, DOI: 10.1016/j.media.2021.102299, 2022.

[27] H. C. Shamsudin, A. Adam, M. I. Shapiai, M. A. M. Basri, Z. Ibrahim and M. Khalid, An improved two-step supervised learning artificial neural network for imbalanced dataset problems, *Proc. of CIM-Sim 2011 3rd Int. Conf. Comput. Intell. Model. Simul.*, vol.8, no.5, pp.108-113, DOI: 10.1109/CIM Sim.2011.28, 2011.

[28] S. Lawrence and C. L. Giles, Overfitting and neural networks: Conjugate gradient and backpropagation, *Proc. of Int. Jt. Conf. Neural Networks*, vol.1, pp.114-119, DOI: 10.1109/ijcnn.2000.857823, 2000.