Reliability Evaluation of Visualization Performance of Convolutional Neural Network Models for Automated Driving

Chenkai Zhang 1) Yuki Okafui 1) Takahiro Wada 1)
1) Ritsumeikan University, Graduate School of information science and engineering
1-1-1 Nojihigashi, Kusatsu, Shiga, 525-8577, Japan, (E-mail: zhang1354558057@gmail.com)

Received on January, 22, 2021

ABSTRACT: As deep learning methods in image recognition have achieved excellent performance, researchers have begun to apply CNNs (convolutional neural networks) to automated driving. However, the explainability for the decision making of automated driving is highly desired. In order to trust the model in automated driving, visualization methods have become important for understanding the internal calculation process of CNNs. Therefore, in a previous study, we proposed a method to evaluate the visualization performance of CNN models by using a mathematical model instead of a human driver to generate a dataset that can determine the ground-truth point in images. However, the reliability of the proposed method for validating the visualization performance was not provided. Therefore, in this paper, we verify the proposed method through two experiments to demonstrate the task-dependent performance and visualization performance during training. The reliability of the visualization performance has been demonstrated through experimental results. Therefore, we proposed an evaluation method for visualization performance in automated driving systems.

KEY WORDS: CNN, Vision Systems, Autonomous Vehicle Navigation [C1]

1. Introduction

In the past decade, automated driving technology has advanced rapidly, mainly due to the progress in deep learning methods. Deep learning methods such as neural networks are one of the key components of self-driving vehicles. However, a problem with these methods is the non-explainability for the decision making of automated operations. They are recommended to provide reasons for their actions so that the passengers can trust the automatic system by understanding what is responsible for the specific behaviors generated by autonomous controllers.

Automated vehicle driving systems can be divided into two categories: perception-planning-action pipeline (1, 2) and end-to-end learning approaches (2). The perception-planning-action pipeline approach has existed before the deep learning methods progressed. This approach depends on detecting human-specified objects, such as road edges and pedestrians, by a controller with if-then rules. Recently, deep learning methods have played a major role in the perception process to recognize the surrounding environment. On the other hand, in the end-to-end systems such as convolutional neural network (CNN) models, the camera image is directly used to calculate the information controlling the vehicle, such as steering/throttle control (2, 3). The end-to-end models generate the output value directly from images, thus they can perform excellent control. However, from the developer/user perspective, these models are not much reliable with regards to their black-box characteristics. Therefore, in order to improve the reliability of end-to-end models, visualization methods that enable humans to understand the internal process of CNN models for automated vehicles become important.

There are many studies aiming to understand deep learning models using visualization methods such as LRP (4), CAM (5), Salient maps (6), VisualBackProp (7), and Zeiler and Fergus model (8). Salient map was proposed by Simonyan et al. (9) and it depends on partial derivatives of the network output with respect to the input for computing the visualization results. It indicates the change of input image that would make the output of classification close to or away from some categories. Zeiler and Fergus (9) proposed the deconvolution method, which runs a backward algorithm that uses the weights at each convolution layer. It propagates the prediction from the output back to the input layer in order to create meaningful patterns in the input space.

These visualization methods have been applied to classification problems. In the classification tasks, the visualization results are associated the pixels in the input image with the output results of the classification. For instance, in the classification of a dog in the image, the regions related to dog in the image have high values. This is a good debugging tool for developers in classification tasks. However, this method is limited. The developers of the CNN models and visualization methods tend to subjectively judge the reliability of the visualization result. This phenomenon is referred to as "the inmates running the asylum." Therefore, researchers have proposed an evaluation method for the visualization results (10, 11); however, so far these methods can only be applied to the classification problems because they need to associate the pixels in the input image with the classification of the output results.

In contrast, in the field of automated driving, most researchers (2, 3) use human-generated data sets to reproduce human behavior because of their excellent driving performance. This is one of the regression tasks, which is different from classification tasks. Visualization methods are also applied to the trained CNN model to visualize the regions in the image responsible for the vehicle controller. However, because of the complexity of human recognition behavior, we cannot accurately compare the visualization result with the regions actually used by human drivers in the input images. For instance, the VisualBackProp method (7) can visualize lane marks in the image using the trained CNN models that trained by human-generated datasets/human driving behavior data, which strongly affect vehicle manipulation. However, due to the lack of understanding of the driver’s recognition behavior, it is unclear whether this result extracts the image areas that the drivers actually used in their operations. Therefore, the visualization results cannot be applied to automated driving systems in order to assess the reliability of the CNN models.
There have been studies \(^\text{(12, 13)}\) that treat human gaze points as the ground-truth point of human recognition behavior. However, because of the importance of peripheral vision, the comparative results between the human gaze and the visualization performance may cause incorrect results in understanding the CNN models.

Therefore, in order to assess the visualization performance of CNN models for automated driving, we proposed a method to use a driver model to generate a dataset instead of using a human-generated data set \(^\text{(14)}\). The previous studies could not strictly determine the regions of the image that humans use for vehicle control, whereas we could determine a specific part of the input image used by the mathematical driver model for the vehicle control. This is a great advantage when we evaluate the visualization performance of CNN models in an automated driving controller because we can compare the decisional point as the ground-truth and visualization results of the CNN model. The proposed method was used to compare the visualization performance of several deep learning networks \(^\text{(14)}\). In addition, based on the comparison results, we built a more reliable neural network architecture. If we can understand how to calculate the steering/throttle control for automated driving from the images, we can trust the CNN models. Thus, demonstrating the visualization performance is important for the extensive use of automated vehicles.

However, the deep learning model is a data-fitting method; thus, regions that are strongly correlated with vehicle operations not limited to the decisional point used by the driver model are also visualized. In other words, we cannot determine if the visualization results really depend on the ground-truth point, which leads to a question about the reliability of the proposed method. If the decisional point does not play a decisive role in the calculation process of CNN models, the visualization result cannot represent the reliability of the CNN model. Therefore, in order to verify the reliability of our visualization method for the CNN models for automated driving, we designed two experiments in this study to show that the CNN model for automated driving actually uses the area around the decisional point. In the first experiment, we compared visualization performance in different tasks. In the second experiment, we validated the visualization performance during the training process of the CNN models for simulating operations using the driver model. In other words, we validated that the visualized area is related to the driving task, and the visualization performance is related to the performance of the CNN model. From these experiments, in this study, we validated the reliability of the proposed method for evaluating the visualization performance of the CNN model for automated driving.

The contributions of this study are as follows:

- This study is an extension of our previous work \(^\text{(14)}\), and we provide experimental validation to demonstrate the reliability of the proposed method for evaluating the visualization performance of the CNN models for automated driving.
- The objective and quantitative measurement of reliable automated systems can be trusted by the general users of the automated driving system.

2. Proposed method

Based on the clear definition of the decisional region in an image, we proposed a method to use the driver model instead of the human to evaluate the visualization performance \(^\text{(14)}\). In this section, we review the driver model for generating datasets and PilotNet architecture as a CNN model. We will also review the evaluation method of the visualization performance in our previous work, which consists of two indicators to objectively and quantitatively compare the decisional point with the visualization results.

\[ \delta = k_n \theta_n + k_t \theta, \]

where \(\theta_n\) represents the angle between the heading angle of the vehicle and the direction to the decisional point, \(\theta_n\) are the differential of \(\theta_n\), \(\delta\) is the differential of the steering angle, and \(k_n\) and \(k_t\) are control gains. This control model can make \(\theta_n\) converge to 0; this means that the automated vehicle achieves tracking control. The vehicle speed is set to a constant value.

![Fig. 1 Overview of the one-point model](image)

This model is a modified version of the well-known “two-point visual control model \(^\text{(15)}\),” which refers to two regions in the far and near for steering control. Our proposed method for evaluating the visualization performance compares this decisional point as the ground-truth point and the visualization results. Considering only a single point as the decisional point makes it simple to assess performance. Therefore, we did not use two points as the decisional point; thus, we call this model the one-point model.

In the virtual environment generated by Vizard 5.0 (WorldViz), the one-point model was used to control the simulated vehicle, and the data from the camera image, as shown in the top panel of Fig. 3 as an example, and steering values were generated.

2.2. PilotNet Architecture

Bojarski et al. \(^\text{(3)}\) proposed a CNN architecture called PilotNet that inputs the camera image and outputs the steering angle. PilotNet consists of nine layers, including a normalization layer, five convolutional layers, and three fully connected layers. The PilotNet training process is shown in Fig. 2. The generated pair of the camera image and the steering angle was used to train PilotNet. Details of the convolution layer and fully connected layer are shown in Tables 1 and 2, respectively.

After PilotNet is trained using a pair of camera images and steering data generated by the one-point model, the predicted steering angle value can be obtained from the road image for automated driving. In addition, the VisualBackProp visualization algorithm can be applied to the trained PilotNet. Then, we can obtain the visualization result in which pixels of the image affects the steering performance (the bottom panel of Fig. 3).
2.3.1. Distance between center point and decisional point

Fig. 4 shows an overview of summarizing each weight in the visualization results into one center point. This center point is used as the representative value of the visualization result to evaluate the visualization performance. Therefore, the distance between the decisional point \((x_d, y_d)\) and the center point \((x_c, y_c)\) represents the error between the feature area and the ground-truth point.

\[
x_c = \frac{\sum_{i=1}^{n} \omega_i x_i}{\sum_{i=1}^{n} \omega_i}, \quad y_c = \frac{\sum_{i=1}^{n} \omega_i y_i}{\sum_{i=1}^{n} \omega_i},
\]

\[
Distance = \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2}.
\]

\(\omega_i\) is calculated by the normalization of pixel brightness in the visualization result, which represents the weight of each non-zero pixel. \(x_c\) and \(y_c\) are the coordinates with non-zero weight values in the image; \(x_d\) and \(y_d\) are the coordinates of the center point calculated from the visualization results; \(x_d\) and \(y_d\) are the coordinates of the decision point in the image.

2.3.2. Divergence of feature distribution

We used the \(Divergence\) indicator to represent the dispersion degree of the visualized distribution. The brighter and the farther the pixels are from the center point, the greater the \(Divergence\). When \(Divergence\) is larger, the feature extraction ability of the neural network is worse.

\[
V_x^2 = \frac{\sum_{i=1}^{n} \omega_i (x_i - x_c)^2}{n}, \quad V_y^2 = \frac{\sum_{i=1}^{n} \omega_i (y_i - y_c)^2}{n},
\]

\[
Divergence = \frac{1}{2} \left( V_x^2 + V_y^2 \right).
\]

\(V_x, V_y\) are the standard deviations of the x-axis and y-axis in the image respectively, and \(Divergence\) as the standard deviation for both directions is calculated by Eq. (5).

3. Experiments and Discussion

In this study, 11 courses are extracted from a real-world map and depicted as having a 3 m width in the virtual environment. The average curvature for each course is shown in Fig. 5. The one-point model is used to track each course and generate steering values in each image through the trials. In this study, the driving speed is set to a constant 90 km/h.

---

*Fig. 2. Flow-process diagram for training PilotNet. The one-point model generates the driving information which can be transformed to a pair of input image and steering angle value for training PilotNet. The trained PilotNet predicts the steering angle from the input images.*

*Fig. 3. Input image (top) and visualized image (bottom).*

*Fig. 4. Pictorial diagram of evaluation method. The white pixels show the weighted pixels in the visualization result. The size of the white circle indicates the brightness of each pixel. The white circle with the red edge indicates the center point summarized by each weighted pixels. The red circle denotes the decisional point used in the one-point model. Distance indicator indicates the distance between the center point and the decisional point.***

---

*Table 1 Details of convolution layers in PilotNet.*

| Convolution layers | Output size | Channel, Kernel, Stride | Activation Function |
|--------------------|-------------|-------------------------|--------------------|
| Conv1              | (97, 29)    | (24, 5, 2)              | ReLu               |
| Conv2              | (47, 13)    | (36, 5, 2)              | ReLu               |
| Conv3              | (22, 5)     | (48, 5, 2)              | ReLu               |
| Conv4              | (20, 3)     | (64, 3, 1)              | ReLu               |
| Conv5              | (18, 1)     | (64, 3, 1)              | None               |

*Table 2 Details of fully-connected layers in PilotNet.*

| Fully-connected layers | (Input size, Output size) |
|------------------------|--------------------------|
| FC1                    | (1152, 100)              |
| FC2                    | (100, 50)                |
| FC3                    | (50, 10)                 |
| Output                 | (10, 1)                  |
Chenkai Zhang et al / International Journal of Automotive Engineering
Vol.12, No.2(2021)

We have proposed an evaluation method \(^{(14)}\) to assess the reliability of the visualization performance by comparing the decisional point as the ground-truth value with the visualization result. In our previous work, the CNN model was trained using the data set generated by the one-point model, and the Adam optimizer was used with a learning rate of 0.001. Training was performed for 32 epochs. The same settings are used in this study.

Deep learning methods are known to have a black-box nature. The main reason is that they are only data-fitting methods; therefore, these models tend to find the correlation between the input image and the output. Thus, the PilotNet model does not necessarily use the pixels around the decisional point. In order to validate the reliability of our evaluation method, in this study, we propose two experiments to demonstrate that the PilotNet model actually uses the pixels around the decisional point to calculate the steering angle.

The first experiment was designed to understand whether the visualization results were task-dependent. We created two datasets for two different tasks: the classification task and the driving task. The classification data set is constructed to train a model that judges whether there are road lines in the input image, as shown in Fig. 6. We denote this classification model as “classification PilotNet.” The driving data set is constructed to train a model that can output the steering angle based on the input image. We denote this model as “driver PilotNet” which is the same as the original PilotNet. By training each PilotNet model in each task, we can compare the visualization results of the PilotNet models for different tasks to validate if the visualization performance is task-dependent.

The second experiment was designed to demonstrate the visualization performance during the training process. In this experiment, we can analyze the relationship between the visualization performance and the prediction performance; in other words, it is necessary to understand how the pixels around the decisional point affect the prediction accuracy of the CNN models. For these two experiments, we created a classification dataset that contained 20,000 images to train a classification model, which can judge if there are road lines in the image. As shown in Fig. 6, 10,000 images contain only the green grassland and blue sky, and the remaining 10,000 images are randomly selected from the driving data set, wherein the images also contain road edges. Another driving data set contained 120,000 images with the steering angle generated by the one-point model for the PilotNet training driver.

After validating the effectiveness of our evaluation method in the second experiment, we will review the two types of architecture for the deep learning model. The first model calculates the current steering angle depending on the image. The second model considers the previous steering angle in addition to the image. The second model \(^{(14)}\) was proposed based on the common sense of human driving.

3.1. Experiment I: Comparison between classification PilotNet and driver PilotNet

In order to show the difference between the visualization results of each PilotNet, we train two PilotNet models using the classification and the driving data set. Fig. 7 shows an example of the differences between the visualization results. The visualized area generated by the classification PilotNet is evenly distributed across all road edge lines. This means that all road edges are needed for the classification tasks. On the other hand, the visualization result of the driver PilotNet model shows that only the middle part of the road edge line is visualized. This indicates that, in order to calculate the steering angle, the driver PilotNet needs only a specific part of the course. The visualization results of the two tasks suggest that the visualized part, especially the pixels of road edge lines around the decisional point, play a decisive role in driver PilotNet.

![Fig. 6 Image without road line (left) and image with road lines (right).](image)

In order to compare the dispersion degree of the two visualization results objectively and quantitatively, we calculated the Divergence indicator of the two models for 11 courses. The Divergence is suitable for evaluating whether the visualization results are distributed on the whole road edge lines. The results of the Divergence indicator for each course are shown in Fig. 8. The course serial numbers are the same as the course serial numbers in Fig. 5. Table 3 shows the average Divergence indicator for all 11 courses.

![Fig. 8 Divergence indicator for classification PilotNet and driver PilotNet.](image)
Table 3 Average Divergence indicators for two CNN models.

| Model          | Divergence |
|----------------|------------|
| Classification| 13.19      |
| Driver PilotNet| 11.18     |

As shown in Fig. 8 and Table 3, the average Divergence indicator is larger than the average results of the driver PilotNet. This is because the feature area in the visualization results of the classification model has a uniform distribution. Furthermore, we can see that the value of the Divergence of driver PilotNet for each course is different, whereas the value of the Divergence of classification models for different courses is almost the same. This indicates that the visualization results of the driver PilotNet are influenced by the key information according to different shapes of road lines, while the classification PilotNet always directly uses all the pixels on the whole road lines. Based on this experimental results, the visualization result of the driver PilotNet is different from the classification model. These results indicate that the visualization performance depends on the tasks.

3.2. Experiment II: Visualization performance during the training process

In our previous work (Zhang et al., 2021), we proposed a novel architecture of the CNN model. This is based on the idea that most driver models (not limited to the two-point model) calculate the discrete increase/decrease steering value $\delta \theta$ related to the current steering angle, rather than the absolute steering value $\theta$. We assumed that the human drivers could correctly perceive the current state, and then the modification for the current steering can be simple and accurate for calculating manipulation. Therefore, for calculating the current steering angle $\delta \theta$, the previous steering angle label $\delta \theta_{t-1}$ is beneficial in the CNN model. Inspired by human driving behavior, we proposed the Boole model, in which Boole is an abbreviation for Based On One Label Early. The model is built to add a step-ahead label to the last fully connected layer FC3 in the original PilotNet. The experimental results (Zhang et al., 2021) suggested that the Boole model has better visualization performance than the original PilotNet model.

In this subsection, we focus on the driver CNN model and the driver CNN model with Boole processing (Boole-driver PilotNet), which are trained by the driving data set. Here we evaluate the relationship between the visualization performance and the performance of the steering angle prediction. In particular, we discuss the relationship between visualization performance and steering angle prediction by examining how visualization performance changes during the training process. We used two types of evaluation phases: the models that were trained and the models that were in the process of being trained. Then, we evaluated the visualization performance as Distance and Divergence for each course in all driving data sets.

Fig. 9 shows the training and validation losses, which are calculated from the mean square error [deg$^2$] between the predicted steering values and the ground-truth steering values. The results show that with the increase in training epochs, the error of the driver PilotNet model in the training and validation set gradually decreases. This result indicates that the model successfully fitted the ground truth steering data and presents a relationship between the input image and the output result. The Boole-driver PilotNet has lower loss in both the training and validation datasets, and it converges to approximately zero within a few epochs.

Fig. 10 shows examples of the visualized images in epochs 0, 2, 15, and 30 for driver PilotNet and Boole-driver PilotNet, respectively. The visualized images of both models in epoch 0 (top) show that all the pixels are randomly visualized because the models are untrained and lack the ability to find the responsible pixels for the calculation. In the second row, the visualized images of both models in epoch 2 show that the visualized pixels successfully converge to the right edge line, but they gather at the bottom right of the image. In the third row, the visualized images of both models in epoch 15 show that the pixels expand to the whole right edge line. In the bottom row, the visualized images of both models in epoch 30 show the visualized pixels converge to the center of the right edge line, and the visualized image of the Boole-driver PilotNet is better than that of the driver PilotNet model because the visualized pixels are closer to the center of the edge line.

Fig. 11 shows the results of the Distance and Divergence indicators as the visualization performance during the training process. As shown in Fig. 11, when both models are not trained at all, such as in epoch 0, the value of the Distance indicator is very
small and the value of Divergence is large. This is attributed to the fact that all pixels in the entire image are randomly visualized at the beginning of training; therefore, the center point calculated according to the feature distribution is located at the center of the image, which is near the decisional point in the one-point model. In the first few epochs, as training progresses in both models, the value of the Divergence indicator decreases, which means that the feature distribution distributed in the entire image converges to a certain point, such as the road lines. However, the Distance indicator increases during this period, which represents the overall position of the feature distribution, is still not ideal. From epoch 2, the Divergence indicator of the feature distribution begins to rise slowly, while the Distance indicator decreases rapidly. This indicates that although the overall distribution scope is slowly expanding, the feature distribution approaches the decisional point.

As shown in Fig. 11, the sharp decline of the Distance indicator suggests that the visualization performance improves during the training process. This means that the pixels around the decisional point gradually play a decisive role in the PilotNet and Boole-driver PilotNet model during the training process. Based on Fig. 10, the prediction loss in the training and validation data sets decreases as well. This suggests that the visualized area helps the driver PilotNet and Boole-driver PilotNet models for better prediction performance. These experiments show that the proposed method correctly represents the reliability of the visualization performance in the CNN models by quantifying the error between the visualization results of the CNN models and the decisional point in images.

Table 4 shows the average indicators for all 11 courses for both models. Fig. 12 and 13 show the Distance and Divergence indicators for both models. The course serial numbers in Fig. 12 and 13 are the same as the course serial numbers in Fig. 5. As shown in Table 4, the Distance and Divergence indicators of Boole-driver PilotNet are both lower than that of driver PilotNet. Accordingly, the Boole-driver PilotNet has better visualization performance by referring to the label in the previous step with the help of a fully connected layer.

### Table 4 Average indicators for two CNN models.

| CNN model            | Distance | Divergence |
|----------------------|----------|------------|
| driver PilotNet      | 30.08    | 11.18      |
| Boole-driver PilotNet| 24.86    | 9.81       |

#### 3.3. Experiment III: Visualization result of complicated road traffic environment

The previous experimental results are in the simple traffic environment which only contains road edge lines. In order to validate the visualization results in the case of a complex traffic environment, we make test images including cars and pedestrians for trained models.

By using the trained driver PilotNet model in Experiment I and II, the visualization results for test images are shown in Fig. 14. The result shows that even other objects appear in the images, the driver PilotNet model, which was trained by the environments with only the road edges and texture, only extracts the pixels of the road edge line for steering angle prediction. This indicates that the driver PilotNet model is task-dependent in that it can extract only the information necessary for driving behavior even in a complex environment. In other words, the evaluation method proposed in this study is assumed to be able to apply in complex environments.

4. Limitations

The purpose of this study was to validate the usefulness to define the human driving model to measure the reliability of CNNs. The experimental results have shown the visualization performance is correctly evaluated in the simple traffic environment. However, the human driver model trained for complicated traffic environments has not been validated. This is because, in this study, we used a simple driver model for a simple environment, but in a complex environment, it may be necessary to use an appropriate driver model (for example, an additional algorithm to avoid pedestrians). This will be our future task to prove the plausibility of our evaluation method in complicated traffic environments.
In addition, all experiments in this study were simulated in a virtual environment to precisely make a dataset that only contains road edge lines. Once we make a valid human driver model for complicated traffic environments in real environments, we will be able to perform practical experiments with real vehicles.

5. Conclusion

The aim of this study was to demonstrate the reliability of our evaluation method for visualization performance in CNN models. This evaluation method uses a driver model to generate a data set, instead of humans. Our previous work presented a novel method to evaluate the visualization performance of CNN models for automated driving. In this study, we perform two experiments to assess the reliability of the visualization performance: Experiment I shows that visualization performance is task-dependent, and Experiment II presents the change in visualization performance during the training process. The experimental results demonstrate the reliability of the proposed method.

This study can be useful for the development of automated driving technology. As explained in the introduction, it is very important for automated driving systems to have explainability, i.e., in automated driving systems that require a high level of safety, it is believed that passengers will feel more comfortable using the automated system if the system explains how the automated vehicle makes decisions and performs control. Therefore, a system that cannot correctly guarantee visualization performance will not be able to provide passengers a sense of security and trust. Therefore, we believe that it is possible to develop a model that gives passengers a sense of security by guaranteeing the explainability of the CNN model, as shown in this study.

This research focuses on a simple environment, that is, only road edges and textures exist in the simulated environment; thus, we only evaluated the visualization performance of an automated driving model in this simple environment. However, the majority of practical models have to deal with much more complex circumstances; thus, a mathematical model used in complex situations should be considered. Therefore, based on the proposed method using the mathematical driver model, we will be able to measure the reliability of the end-to-end driving model in the real environment.

6. Acknowledgments

This research was conducted as a part of the project of JSPS KAKENHI (Grant number 20K14996) and SUZUKI FOUNDATION.

References

(1) S. Grigorescu, B. Trasnea, T. Cocias, G. Macesanu: A survey of deep learning techniques for autonomous driving, Journal of Field Robotics, vol. 37, no. 3, pp. 362-386, (2020).
(2) M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, K. Zieba: End to End Learning for Self-Driving Cars, arXiv preprint, arXiv:1604.07316, (2016).
(3) Z. Yang, Y. Zhang, J. Yu, J. Cai, J. Luo: End-to-end multi-modal multi-task vehicle control for self-driving cars with visual perception, Clinical Orthopaedics and Related Research, (2018).
(4) S. Bach, A. Binder, G. Montavon, F. Klauschen, K. R. Müller, W. Samek: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PLOS ONE, vol. 10, no. 7, e0130140, (2015).
(5) B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba: Learning deep features for discriminative localization, CVPR, (2016).
(6) K. Simonyan, A. Vedaldi, A. Zisserman: Deep inside convolutional networks: visualising image classification models and saliency maps, ICLR Workshop, (2014).
(7) M. Bojarski, A. Choromanska, K. Choromanski, B. Firner, L. D. Jackel, U. Muller, K. Zieba: VisualBackProp: visualizing CNNs for autonomous driving, arXiv preprint, arXiv:1611.05418, (2016).
(8) M. D. Zeiler, R. Fergus: Visualizing and understanding convolutional networks, ECCV, pp. 818-833, (2014).
(9) T. Miller: Explanation in artificial intelligence: insights from the social sciences, Artificial Intelligence, vol. 267, 1-38, (2019).
(10) W. Samek, A. Binder, G. Montavon, S. Bach, G. Montavon, K.-R. Müller: Evaluating the visualization of what a deep neural network has learned, IEEE Transactions on Neural Networks and Learning Systems, (2016).
(11) D. Bau, B. Zhou, A. Khosla, A. Oliva, A. Torralba: Network dissection: quantifying interpretability of deep visual representations, computer vision and pattern recognition (CVPR), (2017).
(12) B. Wolfe, J. Dobres, R. Rosenholtz, B. Reimer: More than the useful field: considering peripheral vision in driving, Applied Ergonomics, vol. 65, pp. 316-325, (2017).
(13) L. Huestegge, A. Bockler: Out of the corner of the driver’s eye: peripheral processing of hazards in static traffic scenes, (2016).
(14) C. Zhang, Y. Okafuji, T. Wada: Evaluation of visualization performance of CNN models using driver model, IEEE/SICE international symposium on system integration, (2021).
(15) D. D. Salvucci, R. Gray: A two-point visual control model of steering, Perception, vol. 33, no. 10, 1233-1248, (2004).