IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 24 April 2023, accepted 12 May 2023, date of publication 17 May 2023, date of current version 24 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3276985

RESEARCH ARTICLE

Surround-View Fisheye Camera Viewpoint Augmentation for Image Semantic Segmentation

JIEUN CHO[1], JONGHYUN LEE[1], JINSU HA[1], PAULO RESENDE[2], (Member, IEEE), BENAZOUZ BRADAÏ[2], (Member, IEEE), AND KICHUN JO[1], (Member, IEEE)

[1]Department of Smart Vehicle Engineering, Konkuk University, Gwangjin-gu, Seoul 05029, Republic of Korea
[2]Driving Assistance Research, Valeo, 94000 Créteil, France

Corresponding author: Kichun Jo (kichun@konkuk.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (MSIT) under Grant RS-2023-00209252, and in part by the Korea Institute for Advancement of Technology (KIAT) funded by the Ministry of Trade, Industry and Energy (MOTIE) (Human Resource Development Program for Industrial Innovation) under Grant P0020536.

ABSTRACT In autonomous vehicles, perception information about the surrounding road environment can be obtained through image semantic segmentation. The fisheye camera commonly used in autonomous vehicle surround view systems offers a wide field of view (FoV), providing comprehensive perception information about the surrounding environment and assisting in understanding complex scenes. However, there is a challenge in model training due to the limited availability of fisheye semantic image datasets, resulting in reduced generalization performance and unreliable results in various test environments. In particular, changes in the position and orientation of the camera result in changes in the camera viewpoint, which can impair the model's segmentation performance. Generally, data scarcity problems are solved using augmentation methods, but existing methods have difficulty reflecting the distortion characteristics of fisheye images. To solve this problem, we propose viewpoint augmentation considering the spatially variant distortion characteristic of fisheye images. First, we use the fisheye camera projection model in reverse to map the captured 2D fisheye image to a point on the surface of a unit sphere in 3D. Then, we change the camera's orientation and position by applying rotation and translation operations to the point. Finally, we reproject the transformed point to the fisheye image to generate a fisheye image with a changed viewpoint. The experimental results show that the proposed augmentation method increases the generalization performance of the model and effectively reduces model performance degradation under changing camera viewpoints, making it suitable for practical applications.

INDEX TERMS Camera viewpoint change, data augmentation, surround-view fisheye camera, image semantic segmentation, intelligent vehicles.

I. INTRODUCTION

Surround-view system of an autonomous vehicle is a camera-based system that provides a 360° view of the vehicle's surroundings, allowing the vehicle to understand its surrounding environment comprehensively. The surround-view system typically uses multiple fisheye cameras mounted outside the vehicle to capture images of the surrounding area. Fisheye cameras use wide-angle lenses with a field of view (FoV) of 180° or greater to capture images that cover a wider area than a pinhole camera. Therefore, this system can provide more comprehensive information to the driver, making it suitable for complex urban environments that require a lot of perception information.

Semantic segmentation can be used to perceive the surrounding environment from this image data. Semantic segmentation is a task of pixel-level classification of an image, allowing for dense and fine tagging of classes. This detailed perception information enables the vehicle to make informed and safe driving decisions, ultimately improving traffic safety. It is typically performed through supervised learning, where a model is trained to minimize the difference between

The associate editor coordinating the review of this manuscript and approving it for publication was Shaohua Wan.

predicted and ground truth segmentations. Supervised learning aims to perform well on any new data within the problem domain based on the patterns learned from the training data. The model's generalization performance can be enhanced by using diverse and abundant labeled data.

By applying semantic segmentation to fisheye cameras, which are a common sensor in most commercial vehicles, a wider field of view can be utilized to acquire more situational awareness information about the surrounding environment. Despite the benefits, limited research has been conducted on fisheye image semantic segmentation due to the scarcity of publicly available fisheye semantic image datasets. Before fisheye datasets become available, there are two main approaches to extracting semantic information from fisheye images. The first approach is rectifying the distorted fisheye images [1]. The rectified images can then be applied to any segmentation solution trained on any available pinhole datasets. However, undistortion on the fisheye image has significant drawbacks, such as information loss at the image edges, resampling distortion, and high consumption of computational resources [2]. Especially losing some FoV contradicts the original intention of using a fisheye camera. The second approach is to generate synthetic datasets by introducing distortion to existing pinhole images such as CityScapes [3], which has the advantage of having relatively more available data than fisheye datasets. However, this approach does not reflect the distortion of real fisheye lenses and has a narrow FoV compared to fisheye cameras [4], [5], [6]. Learning the model directly from real fisheye data and applying semantic algorithms to raw data without undistortion could be an optimal solution. However, to our knowledge, the only high-quality fisheye semantic dataset acquired from the real world is WoodScape [7]. The scarcity of data can degrade the model's generalization ability, which means it could be overfitted to a small number of training data and make incorrect predictions for new data.

The problem becomes even more apparent when the trained model is applied to real-world scenarios. The training and testing environments can differ significantly in many aspects, including weather conditions, time of day, location, and sensor settings. While data for various weather conditions, times of day, and locations are often naturally available during data acquisition, providing data for different sensor settings is rare. Therefore, when the camera setting during testing differs from what was used during training, the model may exhibit reduced generalization ability and make inaccurate predictions. In practice, there can be various reasons for changes in the camera setting. For example, the orientation of the camera may change due to impacts such as aging or accidents in a vehicle. Furthermore, since different vehicles have varying heights and structures, the position where the camera is installed can also differ. Consequently, these differences in the camera's orientation and position can cause a change in the camera's viewpoint, resulting in a degradation in the model's segmentation performance in actual situations.

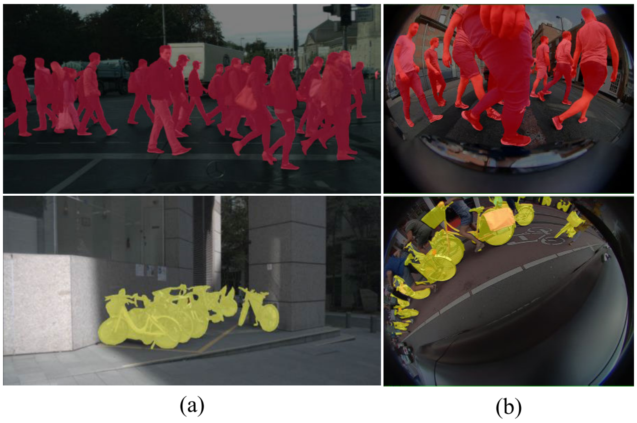

FIGURE 1. Comparison of object shapes in two types of images: (a) pinhole images from the (top) CityScapes [3] and (down) nuImages [9] datasets, and (b) fisheye images from the WoodScape [7] dataset.

The simplest way to address this problem is to create training data from the changed camera viewpoint. However, constructing training data is time-consuming and expensive, starting from data collection to labeling. Therefore, within a limited dataset, data augmentation techniques are generally used to solve the problem of data scarcity [8]. Data augmentation is a technique used to increase the quantity and diversity of the training dataset by applying various transformations to existing image data. By using augmented image data with techniques such as rotation, translation, and scaling, we can increase the robustness of the model to situations where objects can be captured from different viewpoints without additional cost.

However, existing augmentation methods do not consider the distortion characteristics of fisheye cameras. The fisheye images have significant distortion, with the degree of distortion increasing towards the image's periphery. Fig. 1 compares the shape of an object captured with a typical pinhole camera and a fisheye camera. The shape of an object in a pinhole image is similar regardless of which part of the image is observed. In contrast, the shape of an object in a fisheye image varies depending on the object's position in the image. The existing augmentation technique does not alter the shape of an object based on its position within the image, and the shape remains consistent throughout. Therefore, even if the existing augmentation technique is applied, it is difficult for the model to learn individual objects' varying degrees of distortion effectively.

This paper proposes a viewpoint augmentation method that considers the spatially variant distortion characteristic of fisheye images. To our best knowledge, this is the first distortion-aware augmentation method for image semantic segmentation. It can assist in fisheye image semantic segmentation research, which is challenging to train the model due to lack of data, and prevent performance degradation of segmentation models due to changes in camera viewpoint. The overall system consists of three parts, as shown in Fig. 2. The first step (a) is to map all pixels of the 2D image to points on a 3D unit sphere in the camera coordinate system.

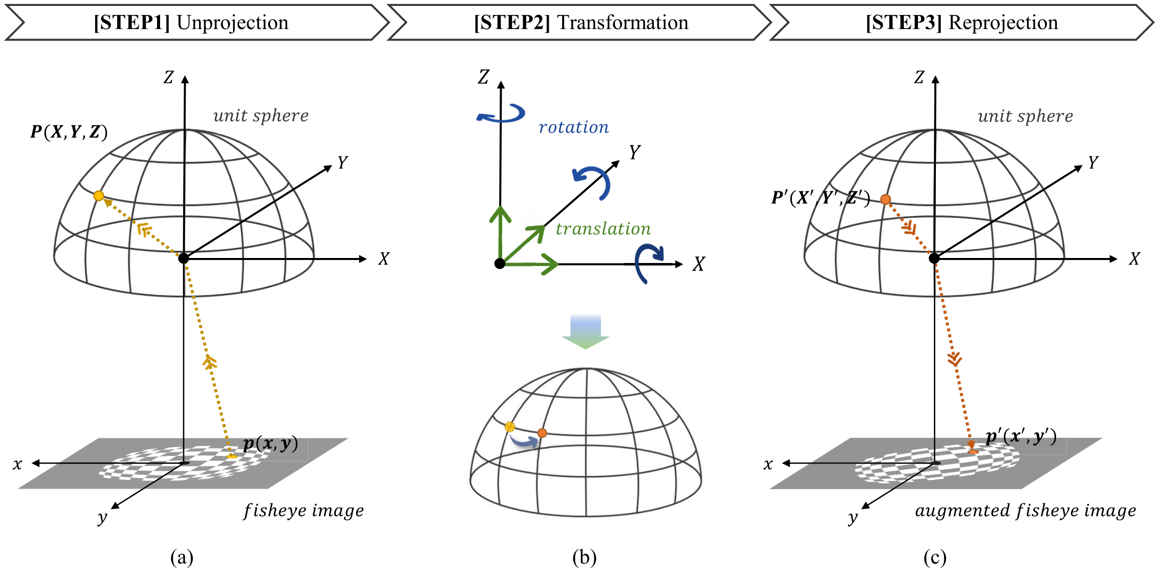

FIGURE 2. System architecture proposed in this paper. The main processes are divided into three parts: (a) 2D image to 3D unit sphere mapping using unprojection, (b) Random rotation and translation of points on a 3D unit sphere and (c) 3D unit sphere to 2D image mapping using reprojection.

TABLE 1. Comparison of image semantic segmentation datasets for autonomous driving acquired in real-world environments.

Camera	Datasets	Year	# of class	# of samples
Pinhole	KITTI	2015	8	0.4k
	CityScapes	2016	30	5k
	Mapillary Vistas	2017	66	25k
	ApolloScape	2018	25	144k
	BDD100k	2018	40	5.7k
	nuImages	2020	23	93k
Fisheye	WoodScape	2021	10	8k

We apply the reverse process of the fisheye imaging principle and achieve mapping to 3D through unprojection. The second step (b) is to randomly rotate and translate the points on the 3D unit sphere with respect to the camera coordinate system. Although this is an operation on the points, it can give the effect of the camera's rotation and movement in the opposite direction. The last step (c) is to reconstruct the transformed points using a fisheye camera projection model into a 2D image. The augmented images are then trained and evaluated in various image semantic segmentation models.

The main contribution of our paper is summarized as follows:

- We analyze the performance degradation of various image semantic segmentation models under diverse view change situations of fisheye cameras using real fisheye datasets.
- We propose viewpoint augmentation using a fisheye camera projection model. It can mitigate performance degradation of image semantic segmentation models in camera view change situations by learning various distortion shapes in which individual objects can be represented in fisheye images.

The paper is organized as follows. In Section II, previous studies are introduced. Section III reviews various camera projection models, and section IV explains the viewpoint augmentation method in detail. Section V describes the evaluation of the proposed method, and we conclude the paper in the last section VI.

II. PREVIOUS STUDIES

A. IMAGE SEMANTIC SEGMENTATION DATASETS FOR AUTONOMOUS DRIVING

The image semantic segmentation datasets can be categorized based on the type of camera used for data acquisition: the Pinhole and the Fisheye camera dataset. Firstly, the **pinhole camera dataset** is composed of narrow FoV pinhole images. Pinhole images have the advantage of having a simple imaging principle and low distortion, but due to the limits of the aperture and image sensor size, they are not easily able to exceed an FoV of 80° [2]. The pinhole cameras are typically used for perceiving distant forward driving environments in autonomous vehicles. KITTI is a pioneering dataset that provides data for various tasks in addition to image semantic segmentation [10]. CityScape provides a large-scale image semantic segmentation dataset that surpasses its predecessors in terms of dataset size and richness of annotation, acquired from 50 cities [3]. Mapillary provides even larger-scale image data acquired from various sensors such as dash cam and smartphone cameras [11]. ApolloScape expands the annotation scale by providing a dataset acquired from high-resolution camera sensors in the driving environments of four cities, with 144K image annotations [12]. BDD100k provides information on 40 classes acquired from driving environments in four cities [13]. NuImages is the latest dataset that provides detailed classification information for vehicles and pedestrians acquired from six surrounding cameras attached to vehicles [9].

On the other hand, the **fisheye camera dataset** consists of fisheye images that capture a wide FoV of 180° or more. Although this introduces significant distortion, it is suitable for obtaining environmental information for the surrounding area within a close range with a small number of cameras, making it mainly used in surround-view systems for autonomous vehicles. However, despite its prevalence, there are few fisheye image semantic segmentation datasets for autonomous driving. WoodScape is the first surround-view fisheye dataset acquired with four cameras mounted outside the vehicle [7]. It provides data for nine tasks: semantic segmentation, monocular depth estimation, 2D and 3D object detection, visual odometry, visual SLAM, motion segmentation, soiling detection, and end-to-end driving.

Table. 1 shows image semantic segmentation datasets for autonomous driving acquired in real-world environments. Pinhole camera semantic segmentation datasets with detailed class labels have been available for a long time. In contrast, there is a significant shortage in quantity and diversity of fisheye image semantic segmentation datasets compared to pinhole cameras, making it challenging for related research. Accordingly, some studies have artificially generated fisheye image datasets from rich pinhole camera datasets for training purposes. Artificial fisheye image is generated based on the formation principle of fisheye images using the geometric distortion model of fisheye lenses. Early studies [4] used the well-known equidistance geometric distortion model to generate data from CityScape [3]. Subsequent studies have generated data with varying degrees of distortion by changing the camera's focal length [5], [14], [15], position and direction [6]. For fisheye data generation for pedestrian detection, projective model transformation (PMT) based on the equidistance distortion model is proposed [16], [17]. Other studies use the projection model of the actually manufactured camera lens introduced in [18] rather than the classical projection model that provides theoretical approximations [19].

B. IMAGE DATA AUGMENTATION

There are two main types of basic image manipulation techniques: Geometric and Photometric transformations. Geometric transformation is the process of changing the overall shape of the entire image by changing the structure in which the pixels that make up the image are arranged. This includes flipping, rotation, translation, cropping, and other techniques. On the other hand, Photometric transformation involves modifying pixel values, such as contrast, sharpness, blurring, brightness, and color changes. When applying the geometric transformation, which is related to pixel position changes, it is necessary to consider the types of the camera. This is because the shapes of objects in fisheye images are spatially variant, unlike pinhole images, where objects appear with similar shapes regardless of their position in the image.

Research on data augmentation considering the characteristics of fisheye cameras can be found in studies on generating artificial fisheye datasets. Some propose zoom augmentation by changing the focal length parameter of the equidistant projection function [4], [5], [14], [15]. This type of augmentation, which takes into account camera characteristics rather than simply scaling images, can generate datasets with varying degrees of distortion depending on the focal length. However, it simulates the distortions in various fisheye cameras that may exist rather than reflecting the distortion of a specific camera. Other studies have made the mapping position of pinhole images movable but only consider left and right movement [17]. Later, six degrees of freedom (6DoF) augmentation for rotation and translation [19] and seven degrees of freedom (7DoF) augmentation including focal length [6] provide better methods for generating various data. Our work is different in that it focuses on data augmentation from fisheye to fisheye, not from pinhole to fisheye. Additionally, it reflects the distortion of an actual lens rather than a geometric projection model designed by assumption.

III. CAMERA PROJECTION MODEL

The camera projection model represents how a camera captures and projects a three-dimensional scene onto a two-dimensional image. Fig. 3 illustrates the pinhole camera projection model and the fisheye camera projection model. A 3D object point $P(X, Y, Z)$ passes through the optical center and reaches a point $p(x, y)$ on the image sensor. The point where the optical axis intersects the image plane is called the principal point $O'(c_x, c_y)$, and the distance between the optical center and the image plane is called the focal length f. The angle between the optical axis and the incident light ray is called the incident angle θ. The projection model expresses the distance ρ between the projected point $p(x, y)$ and the principal point as a function of the incident angle θ.

Pinhole camera projection model assumes that light travels in straight lines and that an inverted image of the external world is projected onto a flat surface through a single point, such as a pinhole or aperture. In other words, a point in the 3D world is projected through the pinhole of the camera and forms an inverted image on the opposite side of the camera sensor. The pinhole projection (also known as perspective projection) is expressed simply using trigonometric ratios, as it assumes that the incident angle and refraction angle are equal:

$$\rho_{pinhole} = f \tan(\theta) \tag{1}$$

On the other hand, the **fisheye camera projection model** describes a fisheye camera's more complex imaging process compared to the simple pinhole camera imaging process. Fisheye cameras use fisheye lenses that provide a much wider FoV than conventional lenses, resulting in circular images with significant distortion around the edges. The fisheye camera projection model maps the 3D world onto a 2D image

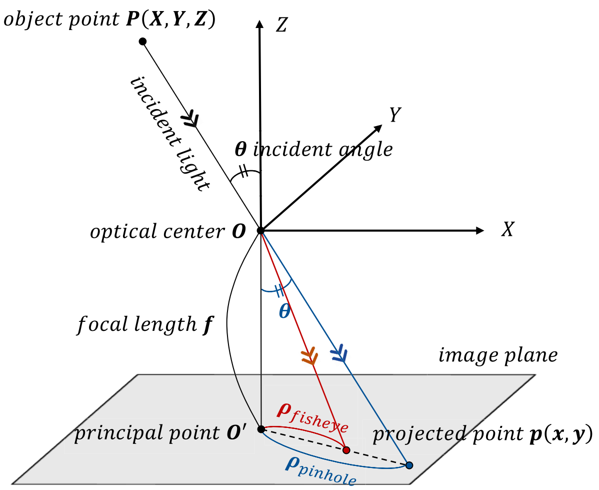

FIGURE 3. Comparison of the imaging principles between a pinhole camera (blue) and a fisheye camera (red).

plane in a non-linear way to explain this distortion. Common types of geometric fisheye projection models include equidistance projection (2) and stereographic projection (3) as in:

$$\rho_{fisheye} = f\theta \tag{2}$$

$$\rho_{fisheye} = 2f\,\tan(\frac{\theta}{2}) \tag{3}$$

The geometric projection models discussed above provide a reasonable approximation of the imaging process, where rays pass through the camera lens and form an image on the image sensor. However, these are simplified models that simulate the non-linear mapping of a real lens and do not accurately represent the lens distortion, sensor misalignment, and manufacturing variations that may affect the image formation process [20]. To accurately represent the imaging process of the real manufactured fisheye camera, applying additional parameters to compensate for deviations from the physical reality [21] or finding out complex mathematical models such as polynomials through the calibration [7] is required.

The WoodScape [7] provides parameters for a more general polynomial model obtained through the calibration:

$$\rho_{fisheye} = k_1\theta + k_2\theta^2 + k_3\theta^3 + k_4\theta^4 \tag{4}$$

When describing the projection model using a polynomial, we can remove the link to the physical property of the lens by changing the focal length as a parameter, and there is no need for additional distortion parameters [2]. In Sec. V-C3, we show the augmentation results based on the difference in the projection functions of the fisheye cameras.

IV. FISHEYE CAMERA VIEWPOINT AUGMENTATION

The proposed viewpoint augmentation method consists of three main steps, as shown in Fig. 4. These steps are based on the fisheye camera projection model mentioned in Sec. III. First, the **unprojection**, which is the reverse process of projection, maps 2D image pixels to 3D points on the unit sphere. Since an image only captures the intensity information of red, green, and blue colors, without any distance information, it is theoretically impossible to obtain the exact distance information of individual pixels from a single image. Therefore, we map all pixels to a unit sphere with a radius of 1. In the context of camera projection, the sphere's radius is only used as a mathematical tool to map 2D image pixels to the 3D surface of the sphere, and it does not have any special meaning. Next, the **transformation** step applies rotation and translation operations to the points on the unit sphere, simulating the camera's moving and rotating effect. Finally, the transformed 3D points are projected back onto the image plane, called **reprojection**. This three-step processing is applied to both RGB and label images, with the only difference being the interpolation method. By varying the parameters of the transformation step, such as the rotation angle and translation distance, we can artificially generate a diverse dataset of fisheye images with varying viewpoints for both RGB and label images.

1) UNPROJECTION

First, we map the pixels in a 2D image to a point on a 3D unit sphere. The image pixels are represented by a pair of discrete integers (u, v) in a u-v coordinate system with the image's top-left corner as the origin. We convert the pixel to an x-y coordinate system with the principal point as the origin and calculate the distance ρ from the origin:

$$x = \frac{u - c_x}{a_x}, y = \frac{v - c_y}{a_y}$$
$$\rho = \sqrt{x^2 + y^2} \tag{5}$$

where (c_x, c_y) is the principal point and (a_x, a_y) is the aspect ratio. The calculated distance value ρ is used to find the real root of θ using a fourth-order polynomial projection model as in (4). The incident angle θ is the angle between the light ray represented as a yellow dotted line and the Z-axis of the camera coordinate system represented as a blue arrow, as shown in Fig. 4(a). Since finding the real root of a fourth-order polynomial for all pixels in an image is time-consuming, we perform the computation in advance for real-time augmentation and store it in a look-up table for reference during model training to quickly retrieve the values. Once we have obtained the value of θ, we can find out the 3D point $P(X, Y, Z)$ on the unit sphere using the following equation:

$$X = \sin(\theta)\cos(\phi) = \sin(\theta)\frac{x}{\rho}$$
$$Y = \sin(\theta)\sin(\phi) = \sin(\theta)\frac{y}{\rho}$$
$$Z = \cos(\theta) \tag{6}$$

where ϕ is the azimuth angle of the 3D point on a sphere with respect to the X-axis of the camera coordinate system represented as the red axis. This angle can be expressed as the angle of the image point $p(x, y)$ with respect to the x-axis of the image coordinate system.

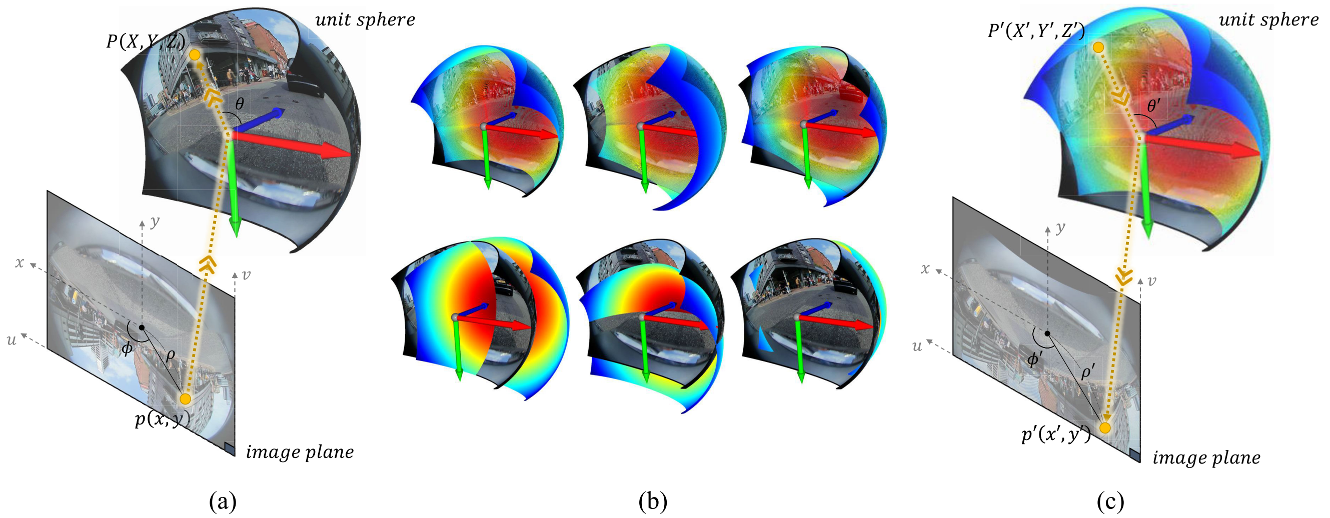

FIGURE 4. Visualization of fisheye camera viewpoint augmentation methods: (a) Unprojection, (b) Transformation, (c) Reprojection. The x, y, and z axes of the camera coordinate system are represented by red, green, and blue arrows, respectively. The yellow circle depicts the movement of image pixels. The rainbow-colored sphere represents the result of applying a transformation to the RGB sphere.

2) TRANSFORMATION

By applying rotation and translation operations to a 3D point $P(X, Y, Z)$ in the unit sphere, a transformed 3D point $P'(X', Y', Z')$ can be obtained. The transformation of 3D points can simulate the effect of rotating and moving the camera in the opposite direction. The equation for this transformation is as follows:

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{7}$$

The transformation equation for a 3D point $P(X, Y, Z)$ involves a 4×4 transformation matrix, which combines the rotation and translation matrices. The rotation matrix $\mathbf{R}$ is a 3×3 matrix representing the point's rotation with respect to the camera coordinate system. It can be described by an angle of rotation and a unit vector representing the axis of rotation. By adjusting the angle and the axis of rotation, various types of rotation can be achieved. The translation matrix $\mathbf{t}$ is a 3×1 matrix representing the point's displacement with respect to the camera coordinate system. It contains the displacement values along the x, y, and z axes, indicating the extent of the point's movement in each dimension. Fig. 4(b) shows the results of applying rotation and translation operations to the 3D points. The original points before the rotation and translation operations are represented by the RGB color that the image had, and the points after the operations are represented by a rainbow-like color. The top three examples show the result of the rotation operation, where the points have rotated by $+20°$ along the X, Y, and Z axes of the camera coordinate system. The bottom three examples show the result of the translation operation, where the points have been translated by $+0.3$ along the X, Y, and Z axes of the camera coordinate system.

3) REPROJECTION

The reprojection process generates an image with a different viewpoint by applying the projection function to the transformed 3D point $P'(X', Y', Z')$. First, the transformed incident angle θ' and distance ρ' are calculated from the transformed 3D point $P'(X', Y', Z')$ using the following equations:

$$\begin{aligned} \theta' &= \arctan(\frac{\sqrt{(X')^2 + (Y')^2}}{Z'}) \\ \rho' &= k_1\theta' + k_2\theta'^2 + k_3\theta'^3 + k_4\theta'^4 \end{aligned} \tag{8}$$

The values of the reprojected image point $p'(x', y')$ are calculated using the following equations:

$$\begin{aligned} x' &= \rho' \cos(\phi') = \rho' \frac{X'}{\sqrt{(X')^2 + (Y')^2}} \\ y' &= \rho' \sin(\phi') = \rho' \frac{Y'}{\sqrt{(X')^2 + (Y')^2}} \end{aligned} \tag{9}$$

where ϕ' is the angle of the reprojected point $p'(x', y')$ with respect to the x-axis of the image coordinate system. It can be expressed as the azimuth angle of the transformed 3D point $P'(X', Y', Z')$. The reprojected image point $p'(x', y')$ is represented as a discrete pixel value in the u-v coordinate system with the top-left corner as the origin, using the following equations:

$$\begin{aligned} u' &= a_x x' + c_x \\ v' &= a_y y' + c_y \end{aligned} \tag{10}$$

Fig. 4(c) depicts the reprojection process. After the transformation, only points with the same azimuth and elevation angle as the original RGB points can be reprojected onto the image, implying that they lie on the same ray. Points that cannot enter the designated area disappear, and the parts that do not intersect within the area are filled with empty values.

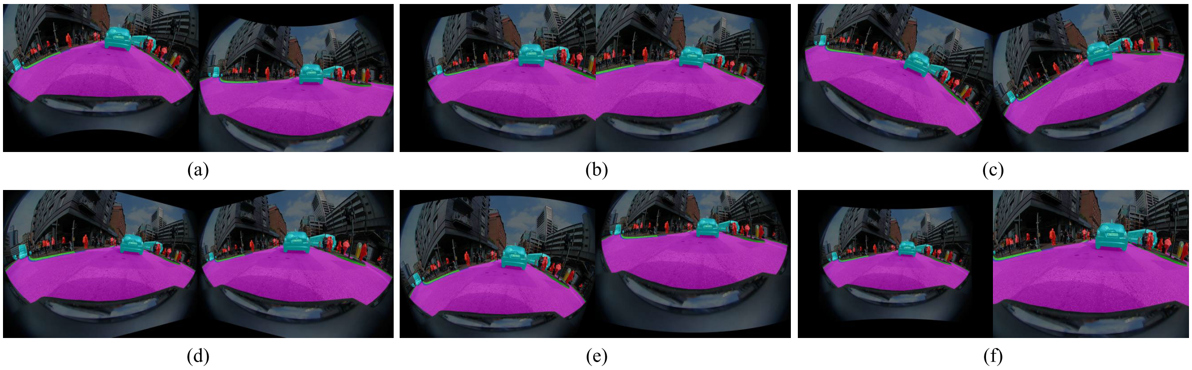

FIGURE 5. Results of applying viewpoint augmentation to the WoodScape [7] dataset. (a)-(c) represent the results of rotating the camera by (left)+20° and (right)−20° along the x, y, and z axes, respectively. (d)-(f) represent the results of moving the camera by (left)+0.3 and (right)-0.3 along the x, y, and z axes, respectively.

Fig. 5 shows the results of the proposed viewpoint augmentation applied to the WoodScape [7] dataset. The image and its corresponding label image are augmented using the same process while employing different interpolation methods. Bilinear interpolation is utilized for RGB images, which fills in the missing pixels by using the pixel values of the four adjacent pixels and their distance ratios. On the other hand, since the label image is a grayscale image that stores label values between 0 and 255 for each pixel, the nearest neighbors interpolation is employed to assign the value of the closest pixel without any changes in label values.

V. EXPERIMENTS

A. EXPERIMENTAL ENVIRONMENT

1) DATASET

We evaluated the effectiveness of our proposed viewpoint augmentation using the WoodScape dataset [7]. WoodScape was collected using saloon vehicles and sports utility vehicles from the United States, Europe, and China. The driving scenarios include highway, urban driving, and parking use cases. The dataset was captured using a fisheye camera with an FoV of 190° from the vehicle's front, rear, left, and right sides. WoodScape provides 8,234 annotated images with a resolution of 1280 × 966 and 23 types of fourth-order polynomial distortion parameters obtained through fisheye lens calibration. The images are randomly split into 4,920 for training, 854 for validation, and 2,460 for testing. The dataset provides 9 classes for semantic segmentation, including road, lane markings, curb, person, rider, vehicles, bicycle, motorcycle, and traffic signs. We trained the model on 10 classes, including the background. Since the WoodScape is not a fully annotated dataset that includes objects such as buildings or the sky, it has a lot of background areas. If the background class is not included in the model training, the model will miss most of the image. Additionally, including the background class enables the model to accurately distinguish the boundaries between the background and the other classes.

2) IMAGE SEGMENTATION MODELS AND TRAINING DETAILS

The experiments were conducted using various popular and contemporary semantic segmentation frameworks. We updated the weights of the pre-trained models with ImageNet using the SGD optimizer and cross-entropy loss for ICNet [22], PSPNet [23], and DeepLabV3+ [24]. For BiseNetV2, we used online hard example mining (OHEM) cross-entropy to calculate the loss and updated the pre-trained model's weights using the SGD optimizer [25]. For SwiftNet, we used the Adam optimizer and cross-entropy loss to update the weights of the pre-trained ResNet18 model with ImageNet [26]. We adjusted the learning rate using different techniques such as a poly learning rate scheduler [22], [23], [24], warm-up poly learning rate scheduler [25], and cosine annealing scheduler [26]. The initial learning rate was determined to be 0.001, which was found to be the most suitable value through our experiments. The batch size for all experiments was set to 4, and we chose the parameters that performed the best on the validation set during 100 epochs as the final model.

3) EVALUATION METRICS

To evaluate the segmentation performance, we used the standard Jaccard index, also known as intersection over union (IoU). The IoU is defined as follows:

$$IoU_n = \frac{TP_n}{TP_n + FP_n + FN_n} \tag{11}$$

where TP_n, FP_n, and FN_n represent the true positive, false positive, and false negative for the nth class. The mean IoU (mIoU), which is the average of IoU for all classes, is formulated as follows:

$$mIoU = \frac{1}{N}\sum_{n=1}^{N} IoU_n \tag{12}$$

where N represents the number of classes.

B. PERFORMANCE DEGRADATION IN VIEWPOINT CHANGE SITUATION

First, we created a test set to investigate the extent of model performance degradation under various camera viewpoint changes. The test set consists of four different scenarios: original images with no changes, as well as small, middle,

TABLE 2. Performance degradation of five benchmark models on various viewpoint change test sets. As the shade of red gets darker, it indicates poorer performance.

Model	Test set	mIoU (%)	degrades (%)	background	road	lanemarks	curb	person	rider	vehicles	bicycle	motorcycle	traffic sign
ICNet	Original	56.95	-	96.6	93.0	65.8	51.2	45.2	39.5	84.0	36.9	33.0	24.3
	Small change	55.14	-3.18	96.1	91.8	64.3	48.9	44.1	36.8	81.2	34.7	31.8	21.7
	Medium change	51.44	-9.68	94.3	88.0	60.9	41.6	38.4	32.6	79.5	34.4	27.1	17.6
	Large change	46.27	-18.75	91.6	80.9	54.8	34.3	35.8	24.9	74.7	28.3	24.3	13.1
PSPNet	Original	66.84	-	97.3	94.1	71.5	58.9	60.7	53.9	88.9	49.0	55.8	38.3
	Small change	64.98	-2.78	97.1	93.8	70.8	57.6	58.7	50.2	87.0	45.7	51.6	37.3
	Medium change	61.56	-7.90	95.9	91.5	67.3	54.2	55.6	46.1	82.2	44.7	44.7	33.4
	Large change	55.08	-17.59	92.8	84.2	61.1	47.2	53.2	38.1	73.4	37.6	36.5	26.7
DeepLabV3+	Original	66.58	-	97.3	94.5	76.8	61.7	58.9	53.3	89.2	46.7	51.3	36.1
	Small change	65.80	-1.17	97.2	94.2	75.8	60.4	57.5	51.8	87.8	46.2	50.6	36.5
	Medium change	63.31	-4.91	96.5	92.7	73.3	56.3	53.4	49.3	85.6	45.7	47.1	33.2
	Large change	58.55	-12.06	95.0	89.1	68.3	49.4	50.6	39.8	81.0	40.3	44.6	27.4
BiseNetV2	Original	63.80	-	97.1	94.0	71.8	59.9	56.3	48.6	87.4	45.2	41.4	36.3
	Small change	62.38	-2.23	96.8	93.4	70.9	58.4	54.8	46.2	85.2	43.7	39.1	35.3
	Medium change	58.43	-8.42	95.0	89.8	65.7	52.8	50.7	42.8	80.9	41.4	34.9	30.3
	Large change	52.23	-18.13	91.7	81.6	58.8	44.8	45.9	32.8	75.2	35.6	33.3	22.6
SwiftNet	Original	64.26	-	97.1	94.1	74.6	58.2	54.8	49.3	86.6	47.0	45.1	35.8
	Small change	63.32	-1.46	96.8	93.7	73.9	57.8	54.4	46.9	84.8	45.9	43.4	35.6
	Medium change	59.96	-6.69	94.9	89.4	68.9	54.5	51.0	44.1	81.3	44.1	38.7	32.7
	Large change	54.79	-14.74	91.6	81.0	63.0	48.4	49.4	37.9	75.0	39.3	37.0	25.3

and large changes in viewpoint. We artificially generated data with changes in camera orientation and position by applying viewpoint augmentation methods to the same test image. For the small change test set, rotations along the x, y, and z axes were set within the range of $[-10°, +10°]$, and the translations along the x, y, and z axes were set within the range of $[-0.1, +0.1]$. The six parameters were randomly determined based on a uniform distribution within the set range. Similarly, the middle change test set had ranges of $[-20°, +20°]$ and $[-0.3, +0.3]$ for rotation and translation, and the large change test set had ranges of $[-30°, +30°]$ and $[-0.5, +0.5]$. By measuring the model's performance on the test set with changes in viewpoint, we can quantify the extent to which the model's performance is degraded.

Table. 2 shows the quantitative results of various image models [22], [23], [24], [25], [26] trained using the WoodScape dataset [7]. No augmentation techniques were applied during training to investigate the impact of viewpoint changes on model performance. The table presents class-wise IoU, mIoU, and the degree of model performance degradation as a percentage. In all segmentation models used in the experiments, the degree of performance degradation increased significantly as camera viewpoint changes became larger. Both class-wise IoU and mIoU consistently showed changes in performance. When the viewpoint change with respect to the training data was relatively small, a small decrease in performance of about 1.17% to 3.18% was observed. On the other hand, when the viewpoint change was large, the segmentation performance decreased significantly by as much as 12.06% to 18.75%. Specifically, we observed a notable decline in segmentation performance for objects such as riders, motorcycles, and traffic signs, which had fewer labeled data than other objects. This performance degradation can be attributed to the limited training data available for these objects, resulting in the model being trained on a restricted set of similar data. As a result, the performance of the model significantly declined when presented with new data, particularly on the test set with varying viewpoints. In real-world scenarios, camera viewpoints can change due to several reasons such as long drives, significant impacts from car accidents, and changes in camera mounting positions. These changes in camera viewpoint can lead to a decline in segmentation performance, making it challenging to trust the perception results. Consequently, there is a need to develop models that can handle viewpoint changes robustly and improve the generalizability of the models to new, unseen data for practical applications.

C. EXPERIMENTAL RESULTS

1) EFFECT OF VIEWPOINT AUGMENTATION

We apply our viewpoint augmentation method to various image segmentation models for training and verify its effectiveness using viewpoint change test sets. Table. 3 shows the experimental results for four cases for each model. The first case is without augmentation, the same as Table. 2. The

TABLE 3. Quantitative evaluation results of five benchmark models. Four different experiments were conducted for each model based on the presence of augmentation. For simplicity, only the mIoU averaged over all classes IoU is presented. The highest score achieved in the original test set is indicated in bold. The values in parentheses represent the percentage decrease in performance compared to the original test set for each model. The color scheme used to depict the degree of performance degradation ranges from dark blue indicating the least decrease to dark red indicating the greatest decrease.

	Augmentation		Test set			
Model	Base	Viewpoint	Original	Small change	Medium change	Large change
ICNet			56.95	55.14 (-3.18%)	51.44 (-9.68%)	46.27 (-18.75%)
	✓		58.73	57.16 (-2.67%)	54.51 (-7.19%)	49.63 (-15.49%)
		✓	59.12	59.27 (+0.25%)	59.03 (-0.15%)	57.60 (-2.57%)
	✓	✓	**60.19**	60.15 (-0.07%)	59.90 (-0.48%)	58.90 (-2.14%)
PSPNet			66.84	64.98 (-2.78%)	61.56 (-7.90%)	55.08 (-17.59%)
	✓		67.70	66.80 (-1.33%)	64.46 (-4.79%)	59.95 (-11.45%)
		✓	**67.77**	67.47 (-0.44%)	67.45 (-0.47%)	66.10 (-2.46%)
	✓	✓	67.73	67.55 (-0.27%)	67.60 (-0.19%)	66.29 (-2.13%)
DeepLabV3+			66.58	65.80 (-1.17%)	63.31 (-4.91%)	58.55 (-12.06%)
	✓		67.33	66.64 (-1.02%)	64.69 (-3.92%)	59.74 (-11.27%)
		✓	67.35	67.25 (-0.15%)	66.83 (-0.77%)	65.27 (-3.09%)
	✓	✓	**67.70**	67.63 (-0.10%)	67.05 (-0.96%)	65.42 (-3.37%)
BiseNetV2			63.80	62.38 (-2.23%)	58.43 (-8.42%)	52.23 (-18.13%)
	✓		65.96	64.35 (-2.44%)	60.36 (-8.49%)	52.33 (-20.66%)
		✓	66.91	66.71 (-0.30%)	66.19 (-1.08%)	64.32 (-3.87%)
	✓	✓	**67.08**	67.03 (-0.07%)	66.56 (-0.78%)	64.49 (-3.86%)
SwiftNet			64.26	63.32 (-1.46%)	59.96 (-6.69%)	54.79 (-14.74%)
	✓		66.00	65.43 (-0.86%)	61.23 (-7.23%)	55.55 (-15.83%)
		✓	67.52	67.56 (+0.06%)	67.10 (-0.62%)	65.46 (-3.05%)
	✓	✓	**68.54**	69.00 (+0.67%)	67.97 (-0.83%)	66.72 (-2.66%)

second case is with base augmentation only, which includes augmentation such as color jittering, gaussian blurring, and horizontal flipping that can be applied regardless of the camera type. Each component of base augmentation is randomly determined. The third case is with viewpoint augmentation only, where we applied our proposed method which includes six parameters related to camera rotation and translation. The rotation for each axis of the camera coordinate system is set to $[-30°, +30°]$, and the translation for each axis is set to $[-0.5, +0.5]$ as the maximum-minimum range. The augmentation is randomly determined for each degree of freedom, and a random value following the uniform distribution within the range set during augmentation is used. The range is experimentally set, considering that the rotation and translation are in the unit sphere. Setting too large a value can cause the augmentation effect to be worse by deforming the image beyond recognition [8]. The fourth case is with both base and viewpoint augmentation. Our proposed augmentation can be performed in real-time during model training, providing the model with diverse inputs using different parameters each time. Applying the base augmentation increases the training time by about 3.3%, while the viewpoint augmentation incurs an additional training time of approximately 6.1%. Applying both augmentations results in a training time increase of approximately 10.3% compared to the case without any augmentation. These additional costs are incurred only during model training and do not affect inference time.

The first column shows the evaluation results on the original test set. The performance of all five semantic segmentation models improved with augmentation applied. When no augmentation techniques were applied, the performance was the lowest. Applying base augmentation showed better performance than not applying any augmentation, and applying viewpoint augmentation showed even better performance. The best performance was achieved when both base and viewpoint augmentations were applied. The performance improvement ranged from 1.33% to 6.66%. This means that the applied augmentation methods provide the model with sufficient and diverse training data by generating different input images every time during the training process, thereby improving the model's generalization performance. The second to fourth columns show the evaluation results on the viewpoint change test sets. Without applying augmentation techniques, there was a significant performance decrease compared to the original performance on the viewpoint change test set. When only base augmentation was applied, it somewhat mitigated the performance degradation, but the effect was minor. Base augmentation techniques such as color jittering, gaussian blurring, and horizontal flipping

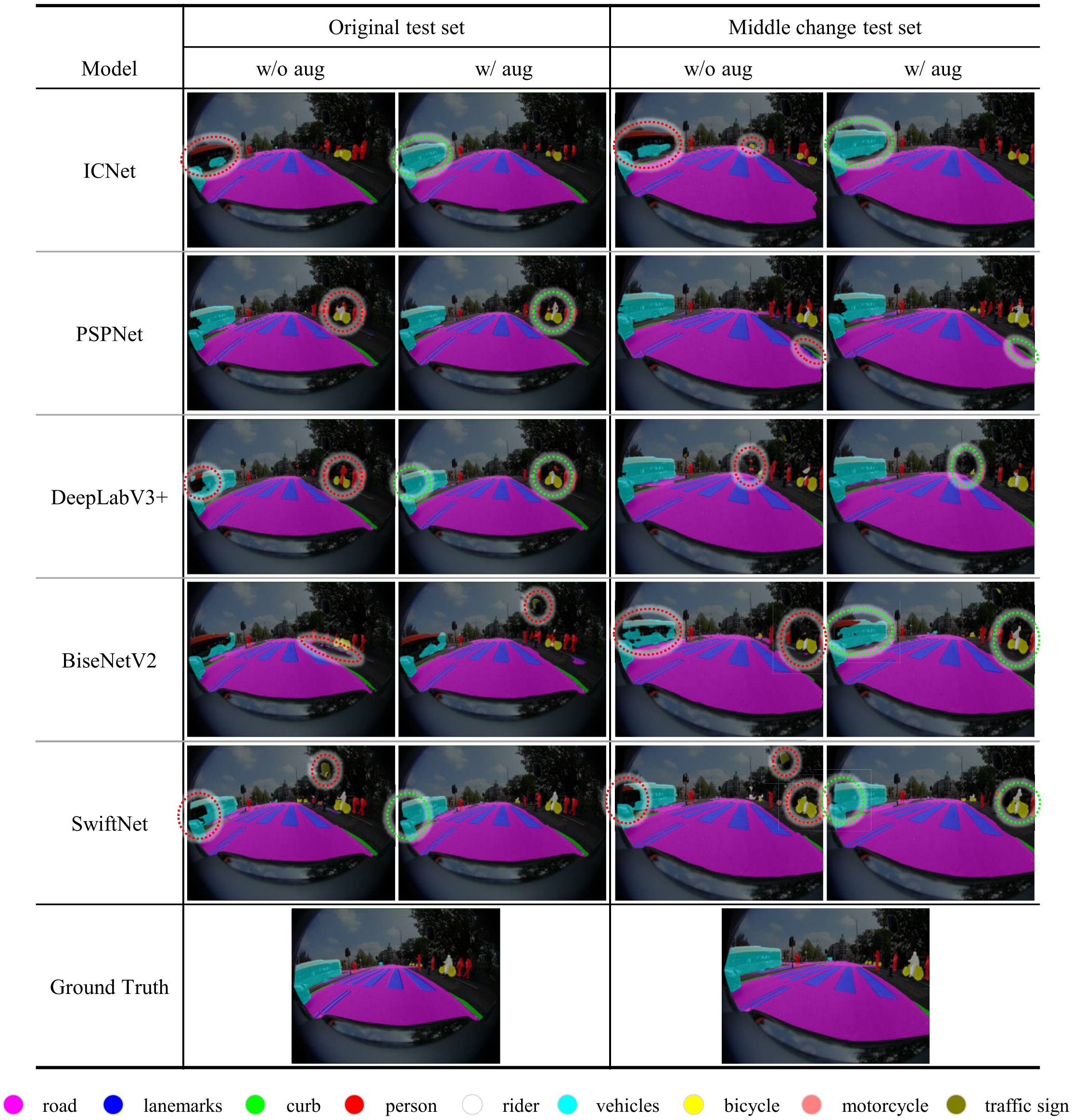

FIGURE 6. Qualitative evaluation results of five benchmark models. Accurately predicted results are represented in the green circle, while inaccurate predictions are represented in the red circle.

are difficult to apply to address changes in camera viewpoints. On the other hand, the third and fourth experimental cases with viewpoint augmentation applied showed a significant reduction in the model's performance decline. The first two experimental cases resulted in a significant drop in performance, up to about 20%, compared to the original. However, in the third and fourth cases with the proposed viewpoint augmentation, the performance degradation was minimal, within 4% relative to the original. In other words, the proposed method not only has a positive impact on the model's generalization performance but also helps the model operate robustly in camera viewpoint change situations.

We qualitatively evaluated the effectiveness of the proposed augmentation methods for five different models, as shown in Fig. 6. The inference results are compared between a model without any augmentations (w/o aug) and a model with both base and viewpoint augmentations applied (w/ aug). For simplicity, we only visualized the results of the original test set and the middle change test set. The models' incorrect predictions are marked red, while correct predictions are green. We can qualitatively verify that applying the viewpoint augmentation techniques generally results in fewer misclassifications and unclassified objects in both test sets.

2) COMPARISON WITH OTHER AUGMENTATION METHOD

This experiment compared the proposed viewpoint augmentation method with conventional augmentation methods such as rotation, translation, and scale. Rotation is rotating an image by a certain angle, usually around the center of the image, and translation is shifting an image's pixels in a horizontal or vertical direction. Scaling is changing the size of an image, either by stretching or compressing it. All three of these transformations can be performed using affine transformation. Each parameter is determined by a random value

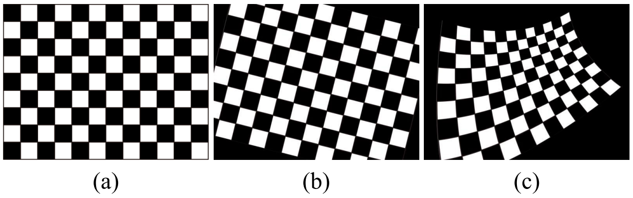

FIGURE 7. The result of applying different augmentation methods. (a) represents the original image, (b) represents the result of affine augmentation, and (c) represents the result of viewpoint augmentation.

TABLE 4. Comparison of results between affine augmentation and viewpoint augmentation. The highest performance for each test set is shown in bold text.

	Test set			
Augmentation	Original	Small change	Medium change	Large change
None	63.80	62.38	58.43	52.23
Affine	66.60	66.18	65.52	62.45
Viewpoint	**66.91**	**66.71**	**66.19**	**64.32**

within the range of $[-30°, 30°]$, $[-0.3, +0.3]$, and $[0.5, 1.5]$, respectively.

Fig. 7 illustrates the difference between the conventional affine augmentation method and the proposed viewpoint augmentation. The conventional method applies simple rotation, translation, and scaling operations to image pixels, maintaining a checkerboard's square shape. In contrast, the proposed method remaps image pixels by considering the distortion of fisheye lenses and camera movements, resulting in varying shapes. Table. 4 shows the results of applying two different augmentation methods to the BiseNetV2 [25] model. Both augmentation methods improve the model's generalization performance and greatly alleviate performance degradation in the camera viewpoint change situation. However, the model with the viewpoint augmentation method performs better on all test sets. Since objects with the same semantic information in fisheye images can appear in various shapes depending on their location, viewpoint augmentation methods that consider the spatially variant distortion characteristic of the fisheye image have demonstrated superior performance.

3) COMPARISON WITH GEOMETRIC PROJECTION MODEL

In this experiment, we analyze the effect of the projection function applied during viewpoint augmentation on the performance of the segmentation model. The fisheye projection function describes a method of mapping a part of a spherical surface onto a flat image. Fig. 8 shows the mapping points on the 3D unit sphere that vary according to the projection function. Fig. 8(a) shows the mapping result using the equidistance projection model as in (2), an ideal geometric model with a linear relationship between the projected radius and the incident angle and is the most commonly used model in fisheye cameras. Fig. 8(b) shows the mapping result using the stereographic projection model as in (3), which maintains the angle, unlike an equidistance model that maintains angular distances. The focal length is determined by the angular coverage and the dimensions of the image, as in:

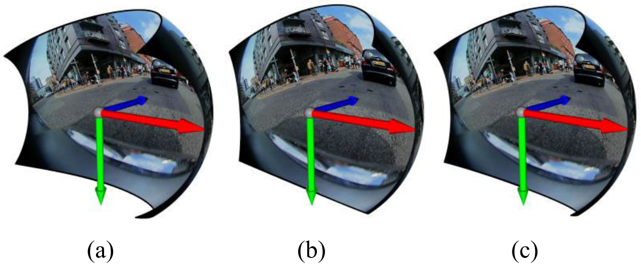

FIGURE 8. Results of unprojection using three different projection functions: (a) equidistance projection model, (b) stereographic projection model, and (c) fourth-order polynomial projection model.

TABLE 5. Comparison of results using three different projection models. The highest performance for each test set is shown in bold text.

	Test set			
Projection	Original	Small change	Medium change	Large change
Equidistance	66.72	66.73	66.14	64.00
Stereographic	66.85	**66.75**	66.12	64.10
4th polynomial	**66.91**	66.71	**66.19**	**64.32**

$$f_{equidistance} = \frac{\rho}{\theta} \tag{13}$$

$$f_{streographic} = \frac{\rho}{2\tan(\frac{\theta}{2})} \tag{14}$$

Fig. 8(c) shows the mapping result of the fourth-order polynomial projection model as in (4) provided in WoodScape [7], which is a mathematical model obtained through calibration. The images are mapped to the unit sphere in different forms, which results in slightly different augmented images. Table. 5 shows the difference in segmentation model performance according to the applied projection function. The fourth-order polynomial projection model yielded the best performance in most test sets, but the difference was negligible. This suggests that a classical geometric model can produce similar results even when an accurate distortion model cannot be obtained through calibration.

VI. CONCLUSION

We proposed viewpoint augmentation for effective learning of fisheye image semantic segmentation. First, we unprojected the image into 3D space on a unit sphere using the fisheye camera projection model. Second, we simulated the change in camera orientation and position by applying transformations to the 3D points on the unit sphere. Third, we generated an augmented image by reprojecting the transformed 3D points back to a 2D image using the fisheye camera projection model. We evaluated the proposed method using the WoodScape dataset and showed significant improvement over existing augmentation methods. In summary,

1) **Generalization performance improvement:** The proposed viewpoint augmentation method provided a diverse set of training data to the segmentation model,

resulting in a 1.33% to 6.66% improvement in generalization performance. This suggested that the data generated by the proposed augmentation method positively impacted model training and could help with fisheye semantic segmentation research that struggled with limited datasets.

2) **Mitigation of performance degradation in camera viewpoint change situations:** The proposed viewpoint augmentation method helped the model learn the individual object's varying degrees of distortion can be represented in fisheye images. As a result, it significantly mitigated the performance degradation in different test sets with viewpoint changes. Without augmentation, there was an up to about 20% performance drop, but with the proposed method, it is reduced within 4%.
3) **The generality of the viewpoint augmentation:** We conducted experiments using accurate projection models obtained through calibration and commonly used geometric projection models. The results showed that the proposed augmentation method positively impacted semantic segmentation regardless of the accuracy of the projection model. This suggested that the proposed augmentation method could be applied generally, even in cases where it was difficult to identify the accurate distortion model.

The proposed method effectively considers the spatially variant distortion characteristics of fisheye images, which leads to improved performance compared to classical augmentation methods commonly used in pinhole images. Although our proposed method successfully simulates changes in viewpoint for individual objects, it has limitations when accurately simulating real-world changes, such as occlusion caused by viewpoint change. Nonetheless, experimental results demonstrate that our proposed method can effectively simulate various degrees of distortion for individual objects, which is a useful capability for many computer vision applications.

In future work, we plan to explore methods for estimating distance values more accurately, allowing us to generate more realistic augmented images [27]. We will also analyze various variables that can occur in test environments beyond changes in camera viewpoint and develop augmentation methods that can robustly handle such changes. Additionally, in practical applications, the low-latency inference is just as important as the robustness of the model to various environments [28]. Therefore, we plan to research reducing network size while minimizing information loss to achieve real-time performance [29].

ACKNOWLEDGMENT

The authors would like to thank Paulo Resende and Benazouz Bradaï for their valuable comments, technical support, and assistance throughout the development of this research.

REFERENCES

[1] R. Varga, A. Costea, H. Florea, I. Giosan, and S. Nedevschi, "Super-sensor for 360-degree environment perception: Point cloud segmentation using image features," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–8.

[2] V. R. Kumar, C. Eising, C. Witt, and S. K. Yogamani, "Surround-view fisheye camera perception for automated driving: Overview, survey & challenges," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 4, pp. 3638–3659, Apr. 2023.

[3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.

[4] L. Deng, M. Yang, Y. Qian, C. Wang, and B. Wang, "CNN based semantic segmentation for urban traffic scenes using fisheye camera," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 231–236.

[5] Á. Sáez, L. M. Bergasa, E. Romeral, E. López, R. Barea, and R. Sanz, "CNN-based fisheye image real-time semantic segmentation," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1039–1044.

[6] Y. Ye, K. Yang, K. Xiang, J. Wang, and K. Wang, "Universal semantic segmentation for fisheye urban driving images," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2020, pp. 648–655.

[7] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, S. Chennupati, M. Uricar, S. Milz, M. Simon, K. Amende, C. Witt, H. Rashed, S. Nayak, S. Mansoor, P. Varley, X. Perrotton, D. Odea, and P. Perez, "WoodScape: A multi-task, multi-camera fisheye dataset for autonomous driving," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9308–9318.

[8] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.

[9] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "NuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11621–11631.

[10] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.

[11] G. Neuhold, T. Ollmann, S. R. Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4990–4999.

[12] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The ApolloScape open dataset for autonomous driving and its application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2702–2719, Oct. 2020.

[13] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2636–2645.

[14] L. Deng, M. Yang, H. Li, T. Li, B. Hu, and C. Wang, "Restricted deformable convolution-based road scene semantic segmentation using surround view cameras," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 10, pp. 4350–4362, Oct. 2020.

[15] Á. Sáez, L. Bergasa, E. López-Guillén, E. Romera, M. Tradacete, C. Gómez-Huélamo, and J. Del Egido, "Real-time semantic segmentation for fisheye urban driving images based on ERFNet," *Sensors*, vol. 19, no. 3, p. 503, Jan. 2019.

[16] Y. Qian, M. Yang, C. Wang, and B. Wang, "Self-adapting part-based pedestrian detection using a fish-eye camera," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 33–38.

[17] Y. Qian, M. Yang, X. Zhao, C. Wang, and B. Wang, "Oriented spatial transformer network for pedestrian detection using fish-eye camera," *IEEE Trans. Multimedia*, vol. 22, no. 2, pp. 421–431, Feb. 2020.

[18] C. Mei, "Laser- augmented omnidirectional vision for 3D localisation and mapping," Ph.D. thesis, INRIA Sophia Antipolis, Project-Team ARobAS, 2007.

[19] G. Blott, M. Takami, and C. Heipke, "Semantic segmentation of fisheye images," in *Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops*, 2018, p. 1–11.

[20] J. Kannala and S. Brandt, "A generic camera calibration method for fisheye lenses," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2004, pp. 10–13.

[21] D. Schneider, E. Schwalbe, and H.-G. Maas, "Validation of geometric models for fisheye lenses," *ISPRS J. Photogramm. Remote Sens.*, vol. 64, no. 3, pp. 259–266, May 2009.

[22] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "ICNet for real-time semantic segmentation on high-resolution images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 405–420.

[23] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2881–2890.

[24] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.

[25] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "BiSeNet v2: Bilateral network with guided aggregation for real-time semantic segmentation," *Int. J. Comput. Vis.*, vol. 129, no. 11, pp. 3051–3068, Nov. 2021.

[26] M. Orsic, I. Kreso, P. Bevandic, and S. Segvic, "In defense of pre-trained ImageNet architectures for real-time semantic segmentation of road-driving images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12607–12616.

[27] V. R. Kumar, S. A. Hiremath, M. Bach, S. Milz, C. Witt, C. Pinard, S. Yogamani, and P. Mäder, "FisheyeDistanceNet: Self-supervised scale-aware distance estimation using monocular fisheye camera for autonomous driving," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 574–581.

[28] Y. Wu, L. Zhang, Z. Gu, H. Lu, and S. Wan, "Edge-AI-driven framework with efficient mobile network design for facial expression recognition," *ACM Trans. Embedded Comput. Syst.*, vol. 22, no. 3, pp. 1–17, May 2023.

[29] C. Chen, C. Wang, B. Liu, C. He, L. Cong, and S. Wan, "Edge intelligence empowered vehicle detection and image segmentation for autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, early access, Jan. 5, 2023, doi: 10.1109/TITS.2022.3232153.

JIEUN CHO received the B.S. degree in smart vehicle engineering from Konkuk University, Seoul, South Korea, in 2022, where she is currently pursuing the master's degree with the Automotive Intelligence Laboratory. Her research interests include deep-learning-based sensor fusion, point cloud semantic segmentation, and domain adaptation.

JONGHYUN LEE is currently pursuing the master's degree with the Automotive Intelligence Laboratory, Konkuk University. His research interests include semantic segmentation, object detection with deep learning using data from camera and LiDAR, and network optimization for intelligence vehicles.

JINSU HA received the B.S. degree in smart vehicle engineering from Konkuk University, Seoul, South Korea, in 2023, where he is currently pursuing the master's degree with the Automotive Intelligence Laboratory. His research interests include applications for LiDAR-based object detection, road condition estimation, and semantic segmentation with deep learning for autonomous vehicles.

PAULO RESENDE (Member, IEEE) received the degree in electrical and computer engineering from the University of Coimbra, Portugal, in 2006. In 2007, he worked with ESRIN, the European Space Agency Center for Earth Observation, Frascati, Italy. In 2008, he was with the Earth Observation and Command and Control Engineering Area of Critical Software, Taveiro, and Lisbon, Portugal. In 2008, he integrated the Project Team IMARA, INRIA Research Center, Rocquencourt, France, where he worked on trajectory planning and control of driverless and cooperative vehicles (cybercars). Since 2014, he has been with Valeo Driving Assistance Research, Bobigny, France, as the Autonomous Driving System Team Leader in the development of a driving assistance research traversable (DART) platform to support the development of advanced driving assistance systems (ADAS) technologies for highly and fully automated vehicles and in particular in the development of Drive4U locate precise localization systems.

BENAZOUZ BRADAÏ (Member, IEEE) received the Ph.D. degree in multisensor fusion from Haute Alsace University, France, in 2007. From 2007 to 2011, he was an Algorithm Engineer and an Expert in ADAS functions, including lighting automation, traffic signs/lights recognition, and eco-driving for hybrid vehicles using cameras and multi sensor fusion with navigation maps. From 2011 to 2018, he was the Project Manager and a Senior Expert in ADAS and automated driving, including the Valeo Urban Automated Driving Drive4U Project. Since 2019, he has been the Innovation Platform Manager of Automated Driving Developments. His role covers the management of transversal/generic automated driving development and several valeo automated driving projects (Cruise4U, Drive4, and eDeliver4U). His research interests include automated driving with several scientific contributions and patents in multi-sensors fusion, precise localization, and mapping and automated driving systems. He is a member of various professional associations, including ADASIS Forum, SAE, and SIA in France.

KICHUN JO (Member, IEEE) received the B.S. degree in mechanical engineering and Ph.D. degree in automotive engineering from Hanyang University, Seoul, South Korea, in 2008 and 2014, respectively. From 2014 to 2015, he was with the ACE Laboratory, Department of Automotive Engineering, Hanyang University, doing research on system design and implementation of autonomous cars. From 2015 to 2018, he was with the Valeo Driving Assistance Research, Bobigny, France, working on the highly automated driving. He is currently an Assistant Professor with the Department of Smart Vehicle Engineering, Konkuk University, Seoul. His current research interests include localization and mapping, objects tracking, information fusion, vehicle state estimation, behavior planning, and vehicle motion control for highly automated vehicles.