Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2024.047695

ARTICLE

A Random Fusion of Mix3D and PolarMix to Improve Semantic Segmentation Performance in 3D Lidar Point Cloud

Bo Liu[1,2], Li Feng[1,*] and Yufeng Chen[3]

[1]School of Computer Science and Engineering, Macao University of Science and Technology, Macao, 999078, China

[2]School of Computer Science and Artificial Intelligence, Chaohu University, Chaohu, 238000, China

[3]Institute of Vehicle Information Control and Network Technology, Hubei University of Automotive Technology, Shiyan, 442002, China

*Corresponding Author: Li Feng. Email: lfeng@must.edu.mo

Received: 14 November 2023 Accepted: 08 January 2024 Published: 16 April 2024

ABSTRACT

This paper focuses on the effective utilization of data augmentation techniques for 3D lidar point clouds to enhance the performance of neural network models. These point clouds, which represent spatial information through a collection of 3D coordinates, have found wide-ranging applications. Data augmentation has emerged as a potent solution to the challenges posed by limited labeled data and the need to enhance model generalization capabilities. Much of the existing research is devoted to crafting novel data augmentation methods specifically for 3D lidar point clouds. However, there has been a lack of focus on making the most of the numerous existing augmentation techniques. Addressing this deficiency, this research investigates the possibility of combining two fundamental data augmentation strategies. The paper introduces PolarMix and Mix3D, two commonly employed augmentation techniques, and presents a new approach, named RandomFusion. Instead of using a fixed or predetermined combination of augmentation methods, RandomFusion randomly chooses one method from a pool of options for each instance or sample. This innovative data augmentation technique randomly augments each point in the point cloud with either PolarMix or Mix3D. The crux of this strategy is the random choice between PolarMix and Mix3D for the augmentation of each point within the point cloud data set. The results of the experiments conducted validate the efficacy of the RandomFusion strategy in enhancing the performance of neural network models for 3D lidar point cloud semantic segmentation tasks. This is achieved without compromising computational efficiency. By examining the potential of merging different augmentation techniques, the research contributes significantly to a more comprehensive understanding of how to utilize existing augmentation methods for 3D lidar point clouds. RandomFusion data augmentation technique offers a simple yet effective method to leverage the diversity of augmentation techniques and boost the robustness of models. The insights gained from this research can pave the way for future work aimed at developing more advanced and efficient data augmentation strategies for 3D lidar point cloud analysis.

KEYWORDS

3D lidar point cloud; data augmentation; RandomFusion; semantic segmentation

1 Introduction

The rapid developments in the field of autonomous driving [1] and 3D detection [2] have fueled the rapid progression of a series of related technologies. These include 3D lidar point cloud technology [3,4], 3D object detection technology for autonomous vehicles [5], object detection and activity recognition in video surveillance [6,7], and offloading technology for mobile edge computing (MEC) based on the Internet of Vehicles (IoV) [8]. A 3D lidar point cloud is a data structure used to represent discrete points in three-dimensional space. It consists of a collection of points, each containing its coordinates in the 3D space. These points can be obtained through techniques such as laser scanning, cameras, or other sensors. Each point in the point cloud has its attributes, commonly including color, intensity, normals, and more. These attributes provide additional information about the objects in the point cloud, such as surface shape, texture, or reflectance properties. 3D lidar point clouds have extensive applications in various fields, particularly in computer vision, robotics, and autonomous driving. They can be used for tasks such as object detection and recognition, semantic segmentation, and so on [3,4]. To process and analyze 3D lidar point cloud data, several preprocessing and feature extraction techniques are typically employed. These may involve operations such as filtering, sampling, or voxelization to reduce noise, decrease data volume, or obtain a regularized representation. Additionally, local features, global features, or descriptors can be computed on the point cloud to facilitate subsequent tasks and algorithms [9,10]. Fig. 1 displays a visualization of a point cloud, which is derived from the 000001 scan of the 03 sequence in the SemanticPOSS dataset [11]. From this perspective, we can observe a traffic road scene where pedestrians are present. Additionally, there are green vegetation and trees on both sides of the road. We can also see parked cars along the roadside and the outlines of buildings.

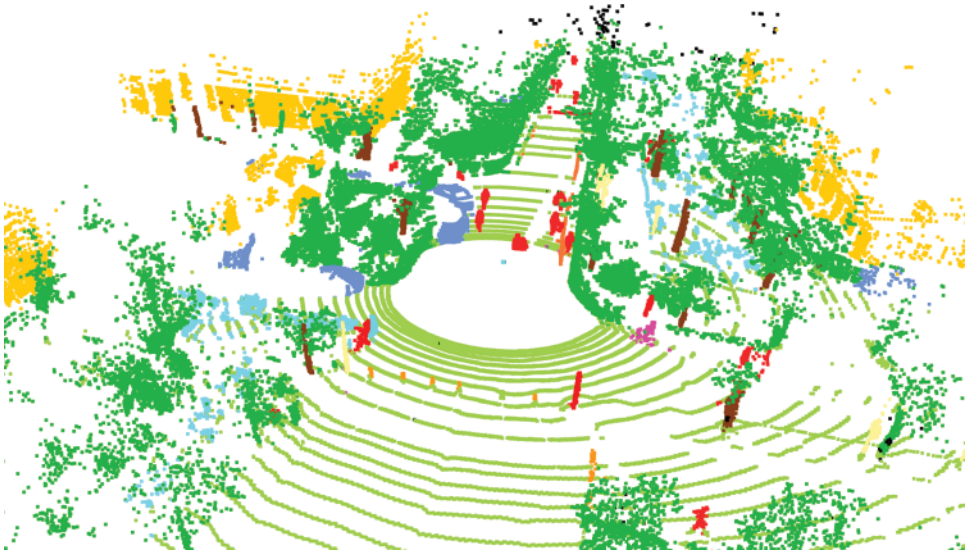

Figure 1: The visualization of a 3D lidar point cloud

Since data collection and labeling is a time and labor-intensive task, it poses a significant challenge for neural networks that require ample training samples, especially in fields where the applications of AI for abnormal detection are class-imbalance, insufficiently labeled fault samples, etc. Therefore, data augmentation plays a crucial role in training deep learning models. It involves generating synthetic variations of the original data by applying geometric transformations, noise injection, or other

operations. These augmented samples introduce additional diversity, which helps the model generalize better and improves its robustness to different real-world scenarios [9,10,12].

There is a wide range of data augmentation methods available for 3D lidar point cloud data. Given the effectiveness of data augmentation for improving neural network models, we conducted an extensive investigation into data augmentation strategies specifically tailored for 3D lidar point clouds. The choice and combination of specific augmentation techniques depend on the characteristics of the dataset and the target task. It is important to strike a balance between introducing sufficient diversity to improve generalization and avoiding excessive distortions that might hinder the model's learning process.

By conducting a comprehensive study on data augmentation strategies for 3D lidar point cloud data, we aim to provide insights into effective techniques that can enhance the performance of neural network models on various tasks involving 3D lidar point clouds. These strategies can contribute to improving the robustness, accuracy, and generalization capabilities of 3D lidar point cloud models in real-world applications. Fig. 2 demonstrates the foundational idea of this research study. The objective is to devise a potent algorithm that can selectively choose one out of N available data augmentation techniques to enhance any given point in a 3D lidar point cloud. In simpler terms, every point in this 3D cloud is augmented using one of these N methods. To break it down further, the far-left box in Fig. 2 symbolizes the diverse data augmentation techniques available for 3D lidar point clouds. The central box signifies the strategic selection of one among these methods. Finally, this chosen method is applied to augment the data point in question.

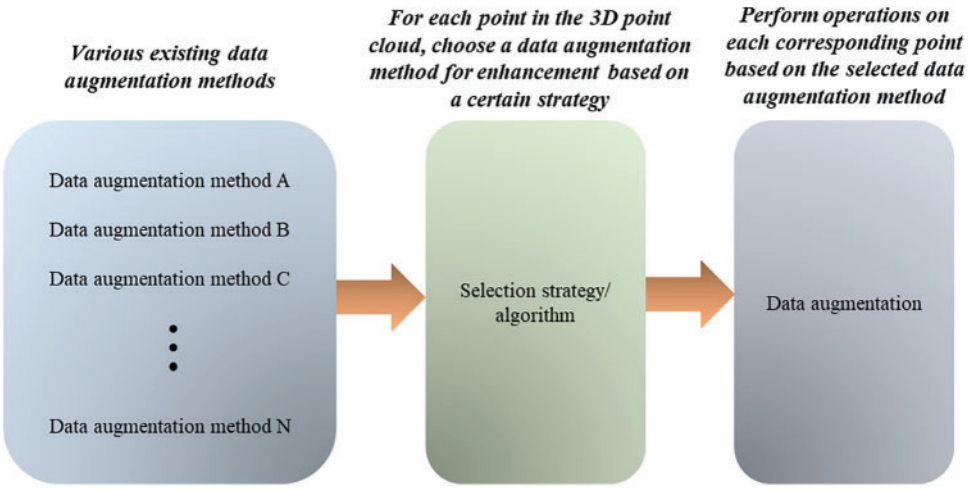

Figure 2: Schematic diagram of RandomFusion data augmentation technique. A, B,..., N represent various data augmentation techniques for the current 3D lidar point cloud. For any point in the point cloud, a selection algorithm is used to choose one of A, B, ..., N for data augmentation

The main contributions of this research are three-fold, with a particular emphasis on the random selection process of data augmentation methods:

• RandomFusion Method: We propose a novel strategy called RandomFusion, which involves the random selection of either PolarMix [9] or Mix3D [10] for augmenting each point within the point cloud data. This approach introduces a flexible and diverse augmentation scheme, allowing for the effective utilization of multiple existing augmentation techniques. This innovative strategy addresses the lack of effective utilization of various augmentation methods in existing research,

opening up possibilities for further exploration and optimization in the field of 3D lidar point cloud data augmentation.

• Performance Improvement: We performed experiments on both the SemanticKitti [12] and SemanticPOSS datasets, the experimental results demonstrate that the RandomFusion method has achieved state-of-the-art performance. Additionally, our strategy did not introduce a noticeable increase in computational complexity.

• Potential for Further Exploration: The random nature of RandomFusion's selection process allows for the integration of other data augmentation methods beyond Mix3D and PolarMix. The approach of randomly selecting one augmentation method from a pool of options provides versatility and potential for researchers to explore and optimize data augmentation strategies for improved performance and adaptability. This flexibility opens up opportunities for researchers to explore hybrid augmentation strategies.

The rest of this paper is organized as follows. In Section 1, we provide an introduction to our research, including a -comprehensive description of our study object–the 3D lidar point clouds, and a detailed overview of our research topic. In Section 2, we delve into the related work and provide a detailed review of prevalent techniques for data augmentation in the context of 3D lidar point clouds. Section 3 is dedicated to a thorough discussion of the proposed random fusion method. We elucidate its principles and provide an in-depth look at its implementation details. In Section 4, we articulate the design and execution of our experimental approach, giving a clear understanding of our methodology. Section 5 presents an extensive analysis and discussion of the results obtained from our experiments, providing valuable insights and observations. Finally, Section 6 concludes this paper.

2 Related Work

Data augmentation techniques have proven to be effective in enhancing the performance and generalization of models in the fields of semantic segmentation and object detection. Within this section, we present a comprehensive survey of data augmentation techniques employed in these tasks, emphasizing notable methodologies and their respective impacts. By drawing inspiration from these methodologies, we aim to propose innovative ideas within the domain of data augmentation.

Traditional data augmentation techniques serve as the foundation for augmenting training data in semantic segmentation and object detection. These techniques typically involve applying random transformations to input data to increase its diversity. Commonly used transformations include random scaling, rotation, translation, flipping, and cropping [13–16]. Spatial transformations are another effective data augmentation technique that applies geometric transformations to images or regions of interest to simulate various viewing angles and perspectives. These transformations help models learn invariant features and enhance their robustness to spatial variations [17].

Generative Adversarial Networks (GANs) have shown promise in generating realistic images and have been leveraged for data augmentation in semantic segmentation and object detection tasks. GAN-based data augmentation methods aim to generate additional training samples by transforming existing images into different styles, viewpoints, or domains. Luc et al. proposed Adversarial Data Augmentation (ADA) for semantic segmentation. ADA employs GANs to generate augmented training samples by learning the data distribution and generating new samples accordingly. The generated samples introduce additional variations and help the model generalize better to unseen data [18]. Gao et al. proposed a new method using numerical simulation and a GAN, the method achieved better performance on detecting gear faults [19]. Lou et al. proposed a fault diagnosis method using

domain adaptation to bridge the gap between simulation signals and measured signals, in which the original simulation fault samples are adjusted using a GAN-based DA network to make them similar to the measured samples through the adversarial training of the refiner and domain discriminator [20]. Isola et al. introduced Pix2Pix, a conditional GAN that learns a mapping from input images to output segmentation masks. Pix2Pix enables the generation of augmented data for semantic segmentation tasks by synthesizing paired images and labels [21]. Huang et al. proposed AugGAN, which leverages GANs to generate augmented training samples for object detection. AugGAN learns the data distribution and generates new samples by transforming existing images into different styles or viewpoints. The generated samples help the model learn robust representations and improve its generalization capabilities [22].

Synthetic data generation is a popular approach for augmenting training data in both semantic segmentation and object detection tasks. Synthetic datasets provide a cost-effective way to generate large amounts of labeled data with diverse variations, which can help improve model performance [23–25]. Weakly supervised data augmentation methods aim to leverage weak annotations or supervision signals to generate additional training samples. These methods help overcome the challenge of obtaining large-scale fully annotated datasets, which are often time-consuming and expensive to create [26,27]. Self-supervised learning has also gained significant attention in recent years as a powerful approach to data augmentation. Self-supervised learning methods aim to learn useful representations by solving pretext tasks on unlabeled data, which can then be transferred to downstream tasks like semantic segmentation and object detection [28].

Style transfer and domain adaptation techniques have been employed to augment training data by transforming images to different styles or adapting them to target domains. In semantic segmentation, Abhinav Valada et al. introduced the AdapNet framework [29], which leverages domain adaptation techniques to transfer knowledge from a labeled source domain to an unlabeled target domain. By adapting the model to the target domain, AdapNet generates augmented training samples that capture domain-specific variations, leading to improved segmentation performance.

Cutout and patch-based augmentation methods involve occluding or replacing parts of an image to encourage the model to focus on relevant features and improve its ability to handle occlusions and partial object appearances. In semantic segmentation, DeVries and Taylor introduced the Cutout technique, which randomly masks out rectangular regions in an image during training. By occluding regions, Cutout encourages the model to learn more robust features and improves its performance in segmenting objects, especially in the presence of occlusion [30]. For object detection, patch-based augmentation techniques have been explored. Chen et al. proposed the GridMask approach, which divides the input image into grids and selectively masks out grid regions during training. GridMask introduces local occlusions and encourages the model to focus on informative regions, improving its ability to detect objects accurately [31].

Mixup and CutMix are data augmentation techniques that involve combining multiple images or patches to create augmented training samples. These techniques encourage the model to learn from mixed samples, enhancing its ability to handle object occlusions, variations, and multi-object interactions. In semantic segmentation, Zhang et al. introduced the Mixup technique, which linearly interpolates pixel-wise labels of two images to generate a mixed image and label [32]. Mixup encourages the model to learn from the mixed samples, improving its generalization to unseen variations and object configurations. For object detection, Yun et al. proposed the CutMix approach, which combines object patches from two images to create a mixed image and label. CutMix encourages the model to learn

from the mixed samples, enhancing its ability to handle object occlusions and improving detection performance [33].

As the research on 3D lidar point clouds progresses, several new methods for data augmentation in 3D lidar point clouds have emerged. These methods largely build upon the principles of data augmentation in the 2D domain, but they also incorporate innovative and customized enhancements to account for the unique characteristics of 3D lidar point cloud data. This combination of leveraging 2D data augmentation concepts while adapting them to suit the specific requirements of 3D lidar point clouds showcases the ongoing exploration and refinement of data augmentation techniques in this field.

PolarMix is another data augmentation method specifically designed for 3D lidar point cloud data, employing the concept of mixing as well. It enhances point cloud distributions and preserves their fidelity through two cross-scan augmentation strategies that involve cutting, editing, and mixing point clouds along the scanning direction. The first step, known as scene-level swapping, entails the exchange of point cloud sectors between two LiDAR scans. These scans are divided along the azimuth axis, allowing for the swapping of corresponding sectors. The second step, referred to as instance-level rotation and paste, involves selecting specific point instances from one LiDAR scan, rotating them at various angles (resulting in multiple copies), and subsequently pasting these rotated instances into other scans [8].

Mix3D, similar to previously mentioned methods like Mixup and CutMix, is a data augmentation technique specifically developed for segmenting large-scale 3D scenes. However, Mix3D incorporates distinct technical details and approaches that set it apart from other methods. It generates novel training examples by combining two original scenes. By exposing objects from a single input scene to the combined context of both mixed scenes, the network learns to disentangle the mixed scene contexts and gains exposure to a wide range of object arrangements that are typically uncommon. Implementation-wise, Mix3D involves concatenating batch entries in pairs. Importantly, the order of points remains unchanged during augmentation, ensuring that the ground truth labels for the mixed point cloud are obtained through concatenation as well [9].

Previous research in the field of point cloud analysis and data augmentation has explored various augmentation techniques and their impact on model performance. However, the random selection process, as proposed in this paper's RandomFusion method, is a novel contribution that enhances the robustness, generalization, and versatility of point cloud analysis models. The integration of different augmentation techniques through random selection opens up new possibilities for further exploration and tailored augmentation strategies.

3 The Proposed Method: RandomFusion

"RandomFusion" can be understood as "random fusion". It refers to the process of randomly combining or merging different elements or methods. In the context of data augmentation, RandomFusion involves randomly selecting and merging various data augmentation techniques or strategies. In some circumstances, this random fusion approach adds an element of variability and unpredictability, as it introduces randomness into the selection process. Instead of using a fixed or predetermined combination of augmentation methods, RandomFusion randomly chooses one method from a pool of options for each instance or sample. This randomization enables the generation of diverse augmented data, enhancing the robustness, generalization, and adaptability of models in various tasks. The random fusion nature of RandomFusion provides flexibility and exploration

potential, allowing researchers to explore different combinations and variations of data augmentation techniques for improved performance and effectiveness.

3.1 Problems to Be Solved

The driving force behind the proposal of the RandomFusion method stems from the proliferation of various data augmentation techniques in point cloud analysis. While numerous methods exist, there is a lack of research on strategies for effectively integrating and utilizing these methods. Many approaches simply stack the augmentation techniques without considering their compatibility, resulting in increased model size, computational complexity, and limited improvement in the effectiveness of data augmentation.

To address this challenge, there is a need for a more intelligent and flexible approach to fuse data augmentation methods. RandomFusion introduces a random selection element where each point in the point cloud is randomly assigned one data augmentation method. This random selection strategy allows the model to choose from multiple methods, providing a more diverse set of augmented samples. Instead of blindly stacking all methods, RandomFusion can yield more significant improvements in data augmentation while maintaining computational efficiency.

By incorporating RandomFusion, we aim to explore superior strategies for data augmentation, moving away from simplistic stacking approaches. With the random selection of different data augmentation methods, we can effectively fuse them to enhance model performance and robustness.

3.2 The Algorithm of RandomFusion

To represent the approach of randomly selecting one augmentation method from a pool of options, we can use a mathematical formulation that involves probability and a set of augmentation methods.

Let us denote the pool of augmentation methods as:

$$M = \{M_1, M_2, M_3, \cdots, M_n\}$$

where each Mi represents a specific augmentation method. In this case, n represents the total number of augmentation methods available in the pool.

To randomly select one augmentation method from the pool, we can assign a probability to each method representing the likelihood of it being chosen. Let's denote the probability of selecting M_i as $P(M_i)$. The probabilities $P(M_i)$ should satisfy the following conditions: they must be non-negative: $P(M_i) \geq 0$ for all i, the sum of probabilities must be equal to 1:

$$\sum P(M_i) = 1$$

where the summation is over all augmentation methods in the pool.

To randomly select an augmentation method, we can use a probability distribution, such as a uniform distribution or a categorical distribution, which assigns equal or custom probabilities to each augmentation method in the pool. For example, in the case of a uniform distribution, where all augmentation methods are equally likely to be selected, the probabilities P(Mi) would be equal for all methods:

$$P(M_i) = \frac{1}{n} \text{ for all } i = 1, 2, \cdots, n.$$

Alternatively, if custom probabilities are desired, they can be assigned based on prior knowledge or experimentation, reflecting the effectiveness or importance of each augmentation method.

The actual process of selecting a specific augmentation method during training can then be formulated as a random selection based on the assigned probabilities. This can be achieved using techniques such as random sampling or using a random number generator to determine the chosen augmentation method according to the assigned probabilities.

In summary, the approach of randomly selecting one augmentation method from a pool of options can be represented mathematically by assigning probabilities to each method and using a random selection process based on these probabilities. The specific choice of probabilities depends on the desired distribution of augmentation methods and can be uniform or customized based on requirements.

Algorithm 1: RandomFusion Method

Input: Point cloud data
Output: Point cloud data after data augmentation
Initialize augmentation methods: $M = \{Mix3D, PolarMix\}$

```
Procedure RandomFusion(data):
    /* Iterate over each point in the data*/
  for each point in data do:
        /* Call a random function for random selection*/
      if random.choice ([True, False])
            /*Apply Mix3D augmentation to the point*/
          point = ApplyMix3D (point)
      else:
            /* Apply PolarMix augmentation to the point*/
          point = ApplyPolarMix (point)
       end if
    end for
     /*Return the enhanced point cloud data*/
    return data
End Procedure
```

Algorithm 1 summarizes the pipeline of the proposed RandomFusion. The pseudo-code demonstrates one of the simplest strategies to randomly select a data augmentation method for each point in a point cloud. The code iterates through each point in the point cloud and randomly chooses between "Mix3D" and "PolarMix" for data augmentation. It employs a loop to iterate through each point. For each point, the algorithm randomly selects between "Mix3D" and "PolarMix" as the data augmentation technique. If "Mix3D" is chosen, the point undergoes "Mix3D" data augmentation. Conversely, if "PolarMix" is chosen, the point undergoes "PolarMix" data augmentation.

The provided pseudocode outlines a procedure called EnhancePointCloud that performs enhancement on point cloud data. The input to the procedure is the point cloud data, and the output is the enhanced point cloud data.

The pseudocode represents the RandomFusion Method, an algorithm designed to augment point cloud data, which are sets of points in a 3D coordinate system. The algorithm initializes two augmentation methods, Mix3D and PolarMix. Within the RandomFusion procedure, it iterates over each point in the input data. For each point, a random function is called to decide which of the two augmentation methods to apply. This is achieved using random.choice ([True, False]), which provides a 50% chance for either choice. If the result is True, the Mix3D augmentation is applied to the point;

if False, the PolarMix augmentation is applied. This procedure is repeated for each point in the data, thereby randomly applying one of the two augmentation methods to each point. Finally, the algorithm returns the augmented point cloud data. The specifics of Mix3D and PolarMix methods are not detailed in this pseudocode but presumably are defined elsewhere in the program.

4 Experiments

4.1 Datasets Preprocessing

The SemanticKitti dataset is a large-scale point cloud dataset designed for semantic segmentation and scene understanding tasks. It is derived from the Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) Vision Benchmark Suite and provides highly detailed semantic annotations for each point in the point clouds.

In the SemanticKitti dataset, the points are annotated with 28 different semantic classes, including car, building, person, vehicle, bicycle, and more. Each point is assigned a class label indicating its semantic class. We adopted the widely practiced approach of using 19 semantic classes from the dataset for evaluation, aligning with other researchers' methodologies.

This dataset was entirely collected from real traffic scenes in Germany. The collection was performed using the Velodyne HDL-64E S3 lidar (Light Detection and Ranging) mounted on the vehicle, which rapidly scans the surrounding environment to generate point cloud data. Each full rotation of the lidar sensor creates a scan. By measuring the time and intensity of the returning laser beams, the lidar obtains the three-dimensional position information of objects in the environment. During the data collection process, the test vehicle drove on German city roads at various speeds and driving modes to capture data from different scenarios. A total of 22 road sequences were collected and labeled as sequence 00 to sequence 21. Following common practices, sequence 08 was used as the validation set, while the training set included sequence 00, 01, 02, 03, 04, 05, 06, 07, 09, and 10.

SemanticPOSS dataset was collected at Peking University and consists of 6 road sequences, labeled as 00 to 05, encompassing a total of 2988 diverse lidar scans in the same data format as SemanticKitti. Following common practices, sequence 03 is used as the validation set, while sequences 00, 01, 02, 04, and 05 are used as the training set. The dataset includes 17 classes in total. In our experiments, we remapped these 17 classes to 14 classes, as shown in Table 1. We excluded the "unlabeled" class in our experiments and focused on calculating the remaining 13 classes.

Table 1: Mapping relationship for remapping the SemanticPOSS dataset from 17 classes to 14 classes

Class numbers (old)	Class numbers (new)	Labels
0	0	1 person
4	0	2+ person
5	1	Rider
6	2	Car
7	3	Trunk
8	4	Plants
9	5	Traffic sign 1 (standing sign)
10	5	Traffic sign 2 (hanging sign)

(Continued)

Table 1 (continued)

Class numbers (old)	Class numbers (new)	Labels
11	5	Traffic sign (high/big hanging sign)
12	6	Pole
13	7	Trashcan
14	8	Building
15	9	Cone/stone
16	10	Fence
17	11	Bike
21	12	Ground
22	23	Unlabeled

According to the algorithm principle of Ploarmix, we need to select some instances and rotate them along the Z-axis by a certain angle before pasting them into the original point cloud [3]. For the SemanticPOSS dataset, our experiments were conducted using the following instance objects: rider, car, trunk, pole, building, and ground. As for the SemanticKitti dataset, we followed the conventional approach in our experiments and selected the following instance objects: car, bicycle, motorcycle, truck, other vehicle, person, bicyclist, and motorcyclist.

4.2 Visualization Settings

To ensure a clear visualization of the experimental results, we have assigned unique colors to each class in the SemanticPOSS dataset, as presented in Table 2. The color assignments align with the official specifications provided by the dataset. These color assignments allow for a visual comparison between the predicted point cloud and the ground truth point cloud during the visualization of the 3D lidar point cloud data. By examining the color of each point, the accuracy of predictions can be intuitively assessed. Taking Fig. 3 as an example, this screenshot is from a scan of a 3D lidar point cloud scene of a campus road at Peking University in the SemanticPOSS dataset. Based on the annotated colors, we can see the green plants, light pink riders, light orange fences, light blue cars, bright red people, orange buildings, yellow poles, and so on.

Table 2: Correspondence between classes and colors in the SemanticPOSS dataset

Class numbers	Colors (RGB)	Colors
0	[255, 30, 30]	Bright Red
1	[255, 40, 200]	Light Pink
2	[100, 150, 245]	Light Blue
3	[135, 60, 0]	Dark Brown
4	[0, 175, 0]	Green
5	[255, 0, 0]	Red
6	[255, 240, 150]	Pale Yellow
7	[50, 255, 255]	Aqua

(Continued)

Table 2 (continued)

Class numbers	Colors (RGB)	Colors
8	[255, 200, 0]	Orange
9	[255, 150, 0]	Orange
10	[255, 120, 50]	Light Orange
11	[100, 230, 245]	Light Blue
12	[150, 240, 80]	Light Green
23	[0, 0, 0]	Black

Figure 3: Illustration of color labels

4.3 The Neural Network Model MinkowskiNet Used for Experiments

MinkowskiNet is a neural network model designed for processing point cloud data. It leverages the Minkowski engine, which utilizes sparse tensors to efficiently handle point cloud data. MinkowskiNet enables the extraction of local and global features from point clouds through its custom convolution and pooling operations. It has been used for tasks such as semantic segmentation, object detection, and point cloud classification. In our experiments, we will utilize the MinkowskiNet model for training, employing the SemanticKitti and SemanticPOSS datasets. Before training, we will apply three data augmentation operations, namely random fusion, polar mix, and Mix3D, to each of these datasets respectively.

4.4 Hyperparameter Setting and Training

According to the algorithm principle of Ploarmix, we need to select some instances and rotate them along the Z-axis by a certain angle before pasting them into the original point cloud [3]. For the SemanticPOSS dataset, our experiments were conducted using the following instance objects:

instance_classes0 = [0, 1, 2,5, 6, 7, 9, 11].

As for the SemanticKitti dataset, we followed the conventional approach in our experiments and selected the following instance objects:

instance_classes = [0, 1, 2, 3, 4, 5, 6, 7].

The training process is performed on an NVIDIA Tesla V100-16G GPU, and the model is trained using Compute Unified Device Architecture (CUDA) version 10.2, Python version 3.8, and PyTorch version 1.6.0. The specific installation command used to install PyTorch and torchvision is "conda install pytorch == 1.6.0 torchvision == 0.7.0 cudatoolkit = 10.2 -c pytorch".

For the SemanticPOSS dataset, the "unlabelled" class has been mapped to class label 23, resulting in a training criterion of cross-entropy loss with an ignore index of 23. This means that during loss computation, predictions corresponding to the index 23 are ignored. In contrast, for SemanticKitti, the "unlabelled" class is conventionally mapped to class label 255, leading to the exclusion of predictions associated with the index 255 during loss calculation.

The training process utilizes stochastic gradient descent (SGD) as the optimizer, with a learning rate of 2.4e-1. To prevent overfitting, a weight decay of 1.0e-4 is applied, and faster convergence is facilitated by a momentum value of 0.9. Nesterov momentum is also employed to accelerate the optimization process.

To adjust the learning rate dynamically, a cosine warmup scheduler is implemented, gradually increasing the learning rate initially and following a cosine annealing schedule. This helps optimize the training process.

5 Experiment Results and Discussion

5.1 Evaluation Metrics

In the context of 3D lidar point clouds, semantic segmentation involves the task of assigning semantic labels to every individual point within the point cloud. The objective is to classify and categorize each point into specific classes, such as objects, surfaces, or regions, based on their semantic meaning or functionality. The outcome of performing semantic segmentation on a 3D lidar point cloud is a labeled point cloud, where each point is colored according to its assigned class label, representing its semantic class. This color-based identification allows for a more intuitive visualization of the segmentation results in our experiments. The purpose of this approach is to provide a clearer understanding of the segmentation outcomes by associating each point with its desired label class through the use of corresponding colors. As shown in Fig. 3, different objects have been identified and colored accordingly. For example, plants and leaves are colored green, tree trunks are colored dark brown, a person is colored bright red while two people or more are colored light pink. Small cars are colored light blue. These color labels in Fig. 3 represent the ground truth, which was manually assigned after human identification of objects. In our experiment, the goal is to predict the class of each point using a neural network model and assign corresponding colors. The colors corresponding to different object instances are presented in Table 2.

In the context of 3D lidar point cloud semantic segmentation tasks, Mean Intersection over Union (mIoU) is a frequently employed metric for assessing the performance of semantic segmentation models. mIoU is determined by calculating the Intersection over Union (IoU) for each class, followed by averaging these IoU values to yield the final performance score. For every class, the IoU is determined by examining the intersection and union areas between the predicted segmentation and the actual, or ground truth, segmentation. This involves comparing the pixels that are assigned to a particular class in both the predicted and ground truth results. The intersection and union areas are then calculated, and the IoU is derived by dividing the intersection area by the union area. Upon calculating the IoU values for all the classes, these values are averaged to obtain the mIoU.

The mIoU value can range from 0 to 1, where 1 signifies perfect segmentation and 0 indicates the worst possible segmentation. A higher mIoU value is indicative of superior segmentation performance across various classes. As such, mIoU serves as a valuable comparative tool for evaluating the performance of different semantic segmentation models, thereby guiding the selection and optimization of these models.

In our experiments, we continue to use mIoU as the primary metric for assessing the quality of experimental results. Additionally, we provide the IoU results for individual classes, which can be found in Tables 3 and 4.

5.2 Experiment Results

We conducted separate sets of experiments on both the SemanticKitti and SemanticPOSS datasets. In Table 3, we present the performance of seven data augmentation methods for semantic segmentation using the MinkowskiNet model and the SemanticKitti dataset. The results for the RandomFusion method are obtained from our experiments, while the results for the other methods are cited from reference [9]. By analyzing the data in Table 3, we observe that the RandomFusion method outperforms all other methods, showing a significant improvement of 2.2% compared to the previously best-performing PolarMix method (increasing from 65% to 67.2%). Moving on to Table 4, it displays the performance of three data augmentation methods, namely RandomFusion, PolarMix, and Mix3D techniques, for semantic segmentation using the MinkowskiNet model and the SemanticPOSS dataset. These results are obtained from our experiments. Similar to Table 3, we find that the RandomFusion method exhibits superior performance compared to the other two methods. It is important to note that while our experiments yielded better results, it does not imply that the RandomFusion method will consistently be effective under all conditions and environments. This variability is a normal occurrence due to the intricacies of deep learning. Therefore, the applicability of these findings to other scenarios requires specific analysis and experimental verification tailored to those particular situations.

To visually demonstrate the advantages of the RandomFusion strategy, we conducted experiments on the SemanticPOSS dataset and visualized the results in Fig. 4. We selected a subset of scans from RandomFusion, PolarMix, Mix3D, and ground truth, capturing them from the same viewpoint. These screenshots are arranged vertically in the order mentioned above. Fig. 4 displays two of these scans, with each column representing a scan, these two scans are from 000001 scan and 000127 scan of 03 sequence in SemanticPOSS.

Based on the visual comparison of segmentation results for tree trunks and stones in the initial scan, it becomes apparent that the RandomFusion method outperforms both PolarMix and Mix3D. As indicated by the red box in the first image of the first column, the superior performance of RandomFusion is visible. When examining the results of the PolarMix method, it is clear that the tree trunk has been inaccurately segmented as plants, and the stone segmentation also appears to be ambiguous. The Mix3D method demonstrates a similar misclassification, incorrectly categorizing the tree trunk as plants. In contrast, the RandomFusion method's segmentation results bear a striking resemblance to the ground truth, depicted in the last row. This close alignment is particularly evident when examining the tree trunk segmentation in the second column, as highlighted by the red boxes. Further examination of the color labels reveals additional disparities among the data augmentation methods. The PolarMix method displays an incomplete segmentation of the two tree trunks, only accurately capturing half of each trunk. In contrast, the Mix3D method falsely segments one of the tree trunks as green plants. On the other hand, the RandomFusion method provides segmentation results that closely mirror the ground truth.From these two sets of visual comparisons, the superior efficacy of RandomFusion in data augmentation is unequivocally demonstrated.

Table 3: Comparison of experimental results based on SemanticKitti dataset

Methods	Car	Bicycle	Motorcycle	Truck	Other-vehicle	Person	Bicyclist	Motorcyclist	Road	Parking	Sidewalk	Other-ground	Building	Fence	Vegetation	Trunk	Terrain	Pole	Traffic-sign	mIoU
MinkNet	95.9	3.7	44.9	53.2	42.1	53.7	68.9	0.0	92.8	43.0	80.0	1.8	90.5	60.0	87.4	64.5	73.3	62.1	43.7	55.9
+CGA	96.3	8.7	52.3	63.2	51.6	63.5	74.4	0.1	93.3	46.6	80.4	0.8	90.3	60.0	88.0	65.1	74.5	62.8	46.8	58.9
+CutMix	96.0	10.2	59.3	78.7	52.1	63.4	79.4	0.0	93.5	47.8	80.7	1.6	90.3	61.0	87.5	66.2	73.3	64.0	46.8	60.6
+CopyPaste	96.6	18.4	62.8	76.3	64.6	68.9	82.8	1.0	93.1	45.3	80.2	1.4	90.5	60.7	88.1	67.8	74.6	63.7	49.1	62.4
+Mix3D	96.3	29.6	61.8	68.5	55.4	72.7	77.7	1.0	94.3	52.9	81.7	0.9	89.1	55.5	88.3	69.3	74.6	65.2	50.3	62.4
+Polar Mix	96.3	51.2	75.6	63.4	63.9	71.9	85.6	4.9	93.6	45.8	81.4	1.4	91.0	62.8	88.4	68.5	75.0	64.6	49.9	65.0
+Random Fusion (ours)	96.8	52.7	75.5	73.9	66.4	74.7	86.2	21.7	94.2	51.6	81.8	0.9	90.7	62.3	88.4	68.5	75.0	64.5	51.0	67.2

Table 4: Comparison of experimental results based on SemanticPOSS dataset

Methods	Person	Rider	Car	Truck	Trunk	Traffic sign	Pole	Trashcan	Building	Cone /stone	Fence	Bike	Ground	mIoU
+Mix3D	62.6	63.3	68.1	44.8	79.2	51.3	38.0	43.6	78.5	37.7	63.1	60.4	81.6	59.4
+PolarMix	61.6	65.6	77.3	33.1	78.5	47.5	41.2	39.0	79.6	42.2	63.3	54.8	80.6	58.8
+Random Fusion (ours)	61.8	64.7	75.8	41.6	77.8	52.3	39.6	41.2	78.8	40.9	64.2	57.6	80.8	59.8

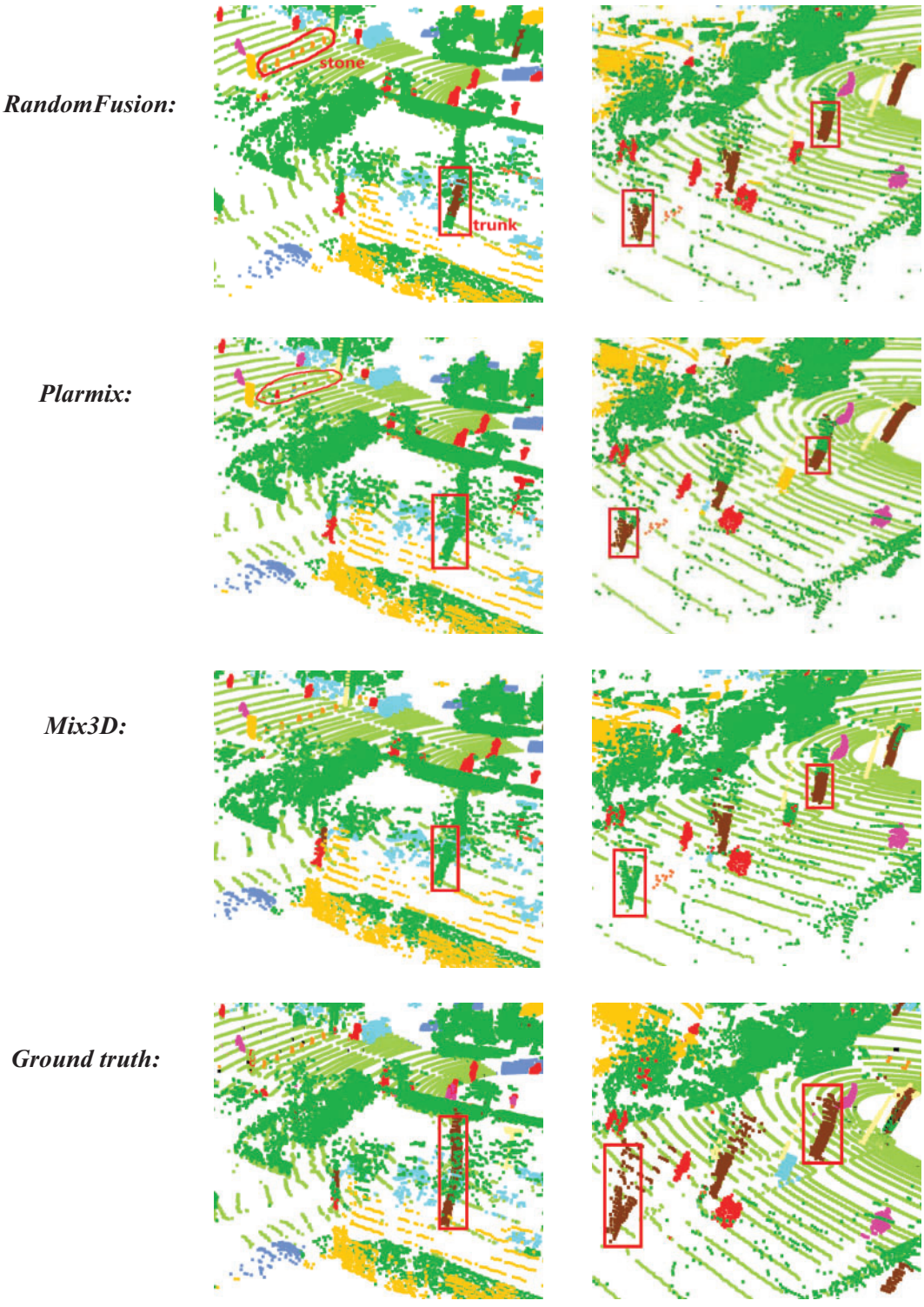

Figure 4: Visualization of comparative experiment results

5.3 Discussion

Based on the ideas presented in this study, we propose three scientific questions that can be further investigated in future research. The first question pertains to the selection problem encountered when integrating various existing data augmentation methods. It involves determining which two or more methods should be chosen for integration. The second question revolves around the strategy employed for the integration process. Should the integration be based on a specific probability distribution or random fusion, or are there alternative integration strategies to consider? The third question involves a labor-intensive task. It entails applying various data augmentation integration strategies to common datasets and evaluating their performance on popular neural network models. By conducting such experiments, it may be possible to create a comprehensive table that specifies the most effective data augmentation integration strategy for a particular dataset and model. This would facilitate the direct application of the findings in practical scenarios. By delving into these three questions, future research may provide valuable insights into the selection and integration of data augmentation methods, ultimately leading to the development of a comprehensive table that guides the application of data augmentation strategies in specific datasets and models.

6 Conclusion

The rapid development of artificial intelligence technology in the field of 3D lidar point clouds has driven the advancement of data augmentation techniques specifically tailored for 3D lidar point clouds. A myriad of methods for augmenting 3D lidar point clouds, including techniques like PolarMix and Mix3D, have emerged in recent years. Nonetheless, the challenge remains in the effective and efficient amalgamation of multiple data augmentation methods, which inhibits the full exploitation of these techniques' potential. This study's primary aim is to unearth superior strategies that facilitate the efficient integration of existing data augmentation techniques. As an initial step, we mesh the PolarMix and Mix3D methods. Precisely, for each point in a 3D lidar point cloud dataset, we employ a random selection process between PolarMix and Mix3D for data augmentation, a technique we term RandomFusion. We undertook comparative experiments on the SemanticKitti and SemantiPOSS datasets, and the results substantiate the efficacy of our proposed RandomFusion method. Moreover, the outcomes illustrate the feasibility of efficiently integrating and harnessing existing data augmentation techniques. Our research serves as a springboard for future investigations into the integration and utilization of existing data augmentation techniques.

Acknowledgement: None.

Funding Statement: This work is funded in part by the Key Project of Nature Science Research for Universities of Anhui Province of China (No. 2022AH051720), in part by the Science and Technology Development Fund, Macau SAR (Grant Nos. 0093/2022/A2, 0076/2022/A2 and 0008/2022/AGJ), and in part by the China University Industry-University-Research Collaborative Innovation Fund (No. 2021FNA04017).

Author Contributions: All authors took part in the discussion of the work described in this paper. Bo Liu proposed the innovative ideas, designed all the experiments, and wrote the main manuscript text. Li Feng is the corresponding author. She takes primary responsibility for communication with the journal during the manuscript submission and publication process. Her contributions also include supervision, investigation, and project administration. Yufeng Chen's contributions include writing-review & editing, funding acquisition, methodology and resources.

Availability of Data and Materials: All the datasets in this paper are from the public data sets on the Internet, which can be easily obtained.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Zhang, Z., Hu, Q., Hou, G., Zhang, S. (2023). A real-time discovery method for vehicle companion via service collaboration. *International Journal of Web Information Systems, 19(5/6),* 263–279.
2. Cao, Z., Xu, L., Chen, D. Z., Gao, H., Wu, J. (2023). A robust shape-aware rib fracture detection and segmentation framework with contrastive learning. *IEEE Transactions on Multimedia, 25,* 1584–1591.
3. Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L. et al. (2020). Deep learning for 3D point clouds: A survey. *IEEE Transactions on Pattern Analysis And Machine Intelligence, 43(12),* 4338–4364.
4. Wu, Y., Wang, Y., Zhang, S., Ogai, H. (2020). Deep 3D object detection networks using LiDAR data: A review. *IEEE Sensors Journal, 21(2),* 1152–1171.
5. Gao, H., Fang, D., Xiao, J., Hussain, W., Kim, J. (2023). CAMRL: A joint method of channel attention and multidimensional regression loss for 3D object detection in automated vehicles. *IEEE Transactions on Intelligent Transportation Systems, 24(8),* 8831–8845.
6. Payghode, V., Goyal, A., Bhan, A., Lyer, S., Dubey, A. (2023). Object detection and activity recognition in video surveillance using neural networks. *International Journal of Web Information Systems, 19,* 123–138.
7. Cao, X., Guo, Y., Yang, W., Luo, X., Xie, S. (2023). Intrinsic feature extraction for unsupervised domain adaptation. *International Journal of Web Information Systems, 19(5/6),* 173–189.
8. Gao, H., Wang, X., Wei, W., AI-Dulaimi, A., Xu, Y. (2024). Com-DDPG: Task offloading based on multiagent reinforcement learning for information-communication-enhanced mobile edge computing in the internet of vehicles. *IEEE Transactions on Vehicular Technology, 73(1),* 348–361.
9. Xiao, A., Huang, J., Guan, D., Cui, K., Lu, S. et al. (2022). PolarMix: A general data augmentation technique for LiDAR point clouds. *Advances in Neural Information Processing Systems, 35,* 11035–11048.
10. Nekrasov, A., Schult, J., Litany, O., Leibe, B., Engelmann, F. (2021). Mix3D: Out-of-context data augmentation for 3D scenes. *2021 International Conference on 3D Vision (3DV)*, pp. 116–125. London, UK, IEEE.
11. Pan, Y., Gao, B., Mei, J., Geng, S., Li, C. et al. (2020). SemanticPOSS A point cloud dataset with large quantity of dynamic instances. *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 687–693. Las Vegas, NV, USA, IEEE.
12. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S. et al. (2019). Semantickitti: A dataset for semantic scene understanding of lidar sequences. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9297–9307. Seoul, Korea (South).
13. Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference*, pp. 234–241. Munich, Germany, Springer International Publishing.
14. Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A. (2017). Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4),* 834–848.
15. Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587. Columbus, OH, USA.

16. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788. Las Vegas, NV, USA.
17. Jaderberg, M., Simonyan, K., Zisserman, A. (2015). Spatial transformer networks. *Proceedings of the 28th International Conference on Neural Information Processing Systems*, vol. 2, pp. 2017–2025.
18. Behpour, S., Kitani, K. M., Ziebart, B. D. (2019). ADA: Adversarial data augmentation for object detection. *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1243–1252. Waikoloa, HI, USA, IEEE. https://doi.org/10.1109/WACV.2019.00137
19. Gao, Y., Liu, X., Xiang, J. (2021). Fault detection in gears using fault samples enlarged by a combination of numerical simulation and a generative adversarial network. *IEEE/ASME Transactions on Mechatronics, 27(5),* 3798–3805.
20. Lou, Y., Kumar, A., Xiang, J. (2022). Machinery fault diagnosis based on domain adaptation to bridge the gap between simulation and measured signals. *IEEE Transactions on Instrumentation and Measurement, 71,* 1–9.
21. Isola, P., Zhu, J. Y., Zhou, T., Efros, A. (2017). Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5967–5976. Honolulu, HI, USA. https://doi.org/10.1109/CVPR.2017.632
22. Huang, S. W., Lin, C. T., Chen, S. P., Wu, Y. Y., Hsu, P. H. et al. (2018). AugGAN Cross domain adaptation with gan-based data augmentation. *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 718–731. Munich, Germany.
23. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M. et al. (2016). The cityscapes dataset for semantic urban scene understanding. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223. Las Vegas, NV, USA.
24. Song, S., Yu, F., Zeng, A., Chang, A., Savva, M. et al. (2017). Semantic scene completion from a single depth image. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1746–1754. Honolulu, HI, USA.
25. Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W. et al. (2017). Learning from simulated and unsupervised images through adversarial training. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2107–2116. Honolulu, HI, USA.
26. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A. (2016). Context encoders: Feature learning by inpainting. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2536–2544. Las Vegas, NV, USA.
27. Dai, J., He, K., Sun, J. (2015). Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1635–1643. Santiago, Chile.
28. Yu, C., Gao, C., Wang, J., Yu, G., Shen, C. et al. (2021). BiSeNet V2 Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision, 129,* 3051–3068.
29. Valada, A., Vertens, J., Dhall, A., Burgard, W. (2017). Adapnet: Adaptive semantic segmentation in adverse environmental conditions. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4644–4651. Singapore.
30. DeVries, T., Taylor, G. W. (2017). Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552.
31. Chen, P., Liu, S., Zhao, H., Jia, J. (2020). Gridmask data augmentation. arXiv preprint arXiv:2001.04086.
32. Zhang, H., Cisse, M., Dauphin, Y. N., Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412.
33. Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J. et al. (2019). CutMix: Regularization strategy to train strong classifiers with localizable features. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6023–6032. Seoul, Korea (South).