EMITTER International Journal of Engineering Technology
Vol. 13, No. 1, June 2025, pp. 22~36
DOI : 10.24003/emitter.v13i1.847

Improving 3D Human Pose Orientation Recognition Through Weight-Voxel Features And 3D CNNs

Moch. Iskandar Riansyah[1,4], Oddy Virgantara Putra[5], Farah Zakiyah Rahmanti[6], Ardiyono Priadi[1], Diah Puspito Wulandari[3], Tri Arief Sardjono[2], Eko Mulyanto Yuniarno[3], Mauridhi Hery Purnomo[3]

[1]Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
[2]Department of Biomedical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
[3]Department of Computer Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
[4]Department of Electrical Engineering, Telkom University, Surabaya Campus, Surabaya, Indonesia
[5]Department of Informatics, Universitas Darussalam Gontor, Ponorogo, Indonesia
[6]Department of Information Technology, Telkom University, Surabaya Campus, Surabaya, Indonesia
Corresponding author: hery@ee.its.ac.id

Received October 4, 2024; Revised April 21, 2025; Accepted May 5, 2025

Abstract

Preprocessing is a widely used process in deep learning applications, and it has been applied in both 2D and 3D computer vision applications. In this research, we propose a preprocessing technique involving weighting to enhance classification performance, incorporated with a 3D CNN architecture. Unlike regular voxel preprocessing, which uses a zero-one (binary) approach, adding weighting incorporates stronger structural information into the voxels. This method is tested with 3D data represented in the form of voxels, followed by weighting preprocessing before entering the core 3D CNN architecture. We evaluate our approach using both public datasets, such as the KITTI dataset, and self-collected 3D human orientation data with four classes. Subsequently, we tested it with five 3D CNN architectures, including VGG16, ResNet50, ResNet50v2, DenseNet121, and VoxNet. Based on experiments conducted with this data, preprocessing with the 3D VGG16 architecture, among the five architectures tested, demonstrates an improvement in accuracy and a reduction in errors in 3D human orientation classification compared to using no preprocessing or other preprocessing methods on the 3D voxel data. The results show that the accuracy and loss in 3D object classification exhibit superior performance compared to specific preprocessing methods, such as binary processing within each voxel.

Keywords: 3D CNN, Weighted, Voxel, Human Orientation, Classification

1. INTRODUCTION

Object classification is one of the critical challenges in object detection within a scene. This scene typically consists of data captured by sensors in the surrounding environment. Object classification studies have primarily focused on image-based computer vision applications. However, the advancement of 3D sensor technologies has shifted the paradigm from 2D to 3D data processing, giving rise to new research opportunities in 3D data processing. 3D data processing finds broad applications in fields such as robotics, autonomous vehicles, 3D medical imaging, military applications, augmented reality, and remote sensing[1–5]. 3D data differs in structure and offers distinct advantages when compared to 2D data. With 3D data, computer vision capabilities are significantly improved as it provides depth information and rich geometric details[6–8]. Although it is possible to convert 2D images into 3D, this often results in errors in perceiving and understanding the surrounding environment, negatively impacting system performance. Additionally, lighting conditions, particularly in fluctuating bright and dark environments, can impact the quality of 2D image data. In contrast, 3D data, with its inherent z-axis component, is more robust. However, the challenge with 3D data lies in its scattered nature, requiring appropriate methods to make sense of the data it forms. Moreover, the vast amount of 3D data necessitates techniques to reduce excessive computational requirements.

One common representation of 3D sensor data is the point cloud. Point clouds are widely favored for 3D data processing in various applications, including robotics, autonomous vehicles, and military domains[3,5]. Other representations include mesh[9] and voxel data[10], which also capture data from 3D sensors[11–13]. However, point cloud data has an extensive footprint and requires decomposition processes to simplify the representation and reduce data size to prevent high computational demands. Voxel data, on the other hand, offers a more structured 3D data format, summarizing large data volumes by grouping sets of points into individual voxels.

2. RELATED WORKS

Research on human orientation estimation has become a focal point of interest for researchers. Several studies have been centered on the development of sensor utilization and the implementation of deep learning algorithms to achieve significant orientation estimation. Various variations in sensor usage, features, and deep learning architectures, especially Convolutional Neural Networks (CNNs), have been employed to enhance robust orientation estimation methods. The use of radar sensors, as observed in this study[14], involves predicting human orientation by monitoring body respiration movement to identify the direction of the human body. While radar can provide rapid estimates, its limitation lies in its inability to provide the necessary detailed information. Therefore, a robust method is required to generate meaningful information.Other studies have attempted to develop methods utilizing wearable devices[15–17] placed on various parts of the

human body. However, this approach may be less flexible as it often requires numerous sensors. One commonly used sensor is the RGB camera[18–20]. Nevertheless, cameras often face challenges related to changes in light intensity and the limitation of spatial information, which is two-dimensional. Hybrid approaches have also been used to achieve improved estimation by combining data from 2D LiDAR sensors and cameras. However, this approach requires significant data and computational resources for processing. An approach that has not been extensively explored is the utilization of a single type of sensor, such as a 3D LiDAR sensor. This sensor possesses vital 3D spatial and geometric elements, with the expectation of achieving more accurate predictions of human body orientation.

3. ORIGINALITY

Various approaches have been explored for estimating human orientation, involving deep learning methods encompassing preprocessing techniques, layer architectures, and the use of relevant datasets. We propose a preprocessing method combined with a CNN architecture to enhance the classification results for several publicly available and frequently used datasets, as well as our primary data. The preprocessing involves weighting on voxelized point cloud data, where each voxel carries a unique weight value to reinforce features before entering the deep learning CNN architecture. The CNN architecture is implemented VGG16, known for its advantages in voxel data classification, as indicated in previous research on orientation prediction comparing various CNN architectures[21] as seen in Figure 1.

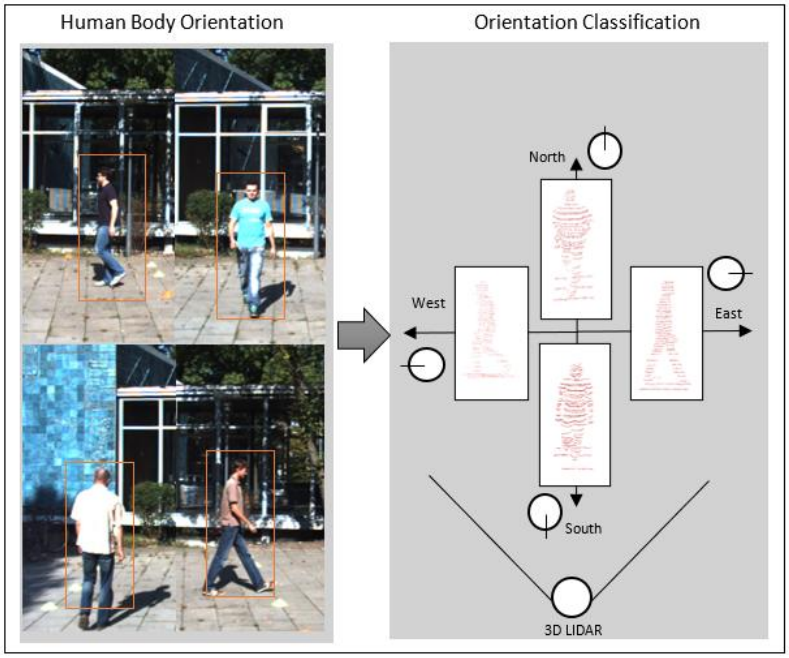


Figure 1. Estimating the direction or motion orientation of a human using 3D LiDAR data

The weighting approach draws inspiration from previous research[22] used in the 3D object reconstruction process from point cloud data. Our study on 3D human orientation estimation is related to classification since we divide the dataset into various orientation classes based on point cloud data. Based on several references, classification can indeed be divided into two categories: discrete classification and continuous regression for 3D human orientation estimation[23].

4. SYSTEM DESIGN

The proposed method in this paper is a combination of preprocessing techniques that apply weighting to 3D voxel objects and process them for classification using deep learning and 3D CNN. The initial input data is a 3D object in the form of a point cloud, which will be converted into voxel format with dimensions of 16x16x16, a process commonly known as 3D data voxelization in Figure 2. The voxelization is not represented as binary values as commonly done but instead as weighted values.

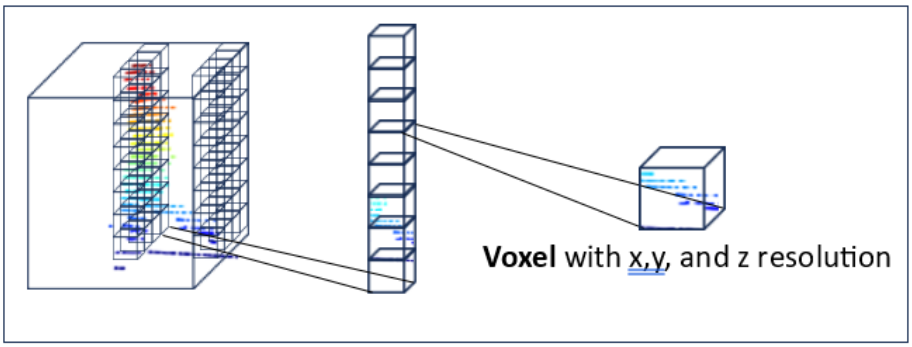


Figure 2. Representation of 3D Human object in voxel form

4.1 Weighted Voxel

Unlike other voxels, which take the value 0 for parts containing points below a threshold or 1 for voxels containing many points above the threshold, the weighted method was introduced in previous research on 3D object reconstruction[22], forming the voxel representation values as integers. In conventional voxels, each voxel is independent of the others, leading to a loss of structural information. In this weighting approach, the final reconstruction from the voxelization process is used to address the issue of inter-neighbor voxel information relationships, also contributing to the formation of new features based on voxel relationships. To achieve this, we employ a 3 × 3 × 3 filter convolved over the conventional binary voxels as seen the following Figure 3.

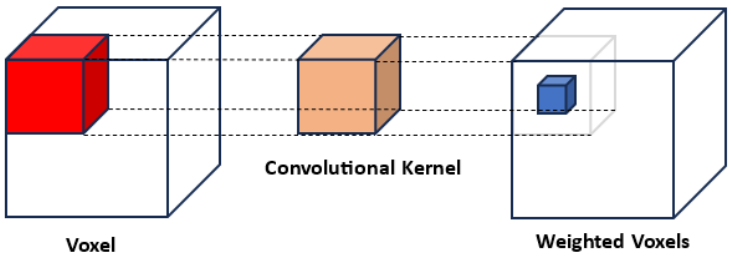


Figure 3. The process of generating weighted voxels involves convolution using a 3x3x3 kernel or filter on the binary- voxels

Based on Figure 3, the calculation of voxel weight values can be performed using the following equation (1).

$$y_{(i,j,k)} = -\omega(-1)^{v_{(i,j,k)}} - \sum_{m=i-1}^{i+1}\sum_{n=j-1}^{j+1}\sum_{p=k-1}^{k+1}(-1)^{v_{(m,n,p)}} \quad (1)$$

Where $v_{(i,j,k)} \in \{0,1\}$ is denoted as the value of conventional voxels, and ω is set to a value of 26. Specifically, we define that voxels with a value of zero are transformed into negative values in Weighted Voxels. Conversely, positive values in Weighted Voxels are derived from voxels with a value of one in conventional voxels. Furthermore, higher values in Weighted Voxels indicate higher density within the 3D object, while lower values denote lower density. Compared to conventional voxels, Weighted Voxels provide richer information, facilitating subsequent reconstruction steps. Set $v_{(i,j,k)}| = 0$ when $i = -1, j = -1, k = -1$.

4.2 3D Convolution

3D convolution is a step to extract features from both spatial and temporal information. 3D convolution is performed by convolving a cube-shaped 3D kernel, combining several adjacent frames. With this construction, new feature maps are generated when the convolution process is connected to several consecutive frames in the previous layer, capturing information from the 3-dimensional movement of the kernel. The equation for 3D convolution with values at position (x, y, z) in feature map -j in the i-th frame layer is given by.

$$v_{ij}^{xyz} = tanh\left(b_{ij} + \sum_m \sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1}\sum_{r=0}^{R_i-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)}\right) \quad (2)$$

$R_i$is denoted as the size of the 3D kernel in the temporal dimension, and w_{ijm}^{pqr} is the value (p, q, r) of the kernel connected to feature map k in the previous layer. An illustration of 3D convolution is provided in Figure 4. It should be noted that the 3D convolution kernel can only extract one type of feature from the frame cube because the kernel weights are replicated across the entire cube. A typical design principle in CNNs is that the number of feature

maps should increase in the later layers by extracting various types of features from the same set of lower-level feature maps.

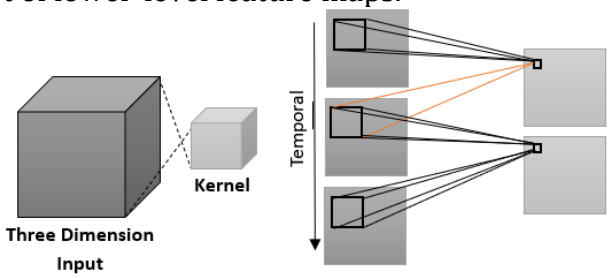


Figure 4. 3D Convolution Step Process

4.3 3D Convolutional Neural Network Architecture

CNN has various variants that have been developed based on previous research studies. The proposed method employs the VGG16 architecture, referring to a study[21] that demonstrated VGG16's superior performance compared to other architectures, such as Resnet and DenseNet, in the orientation classification process using regular binary voxel data, containing values 0 and 1. This served as the basis for our combination with preprocessing that utilizes weighting on voxel data to enhance the classification performance, which was not present in the pure data voxel used in the previous research. However, VGG16 is typically designed and used for 2D data. In this case, we have modified it to take the form of a 3D CNN and have adjusted some parameters in the fully connected layer, which differ from the original VGG16, as shown at Figure 5. These modifications are shown in Table 1.

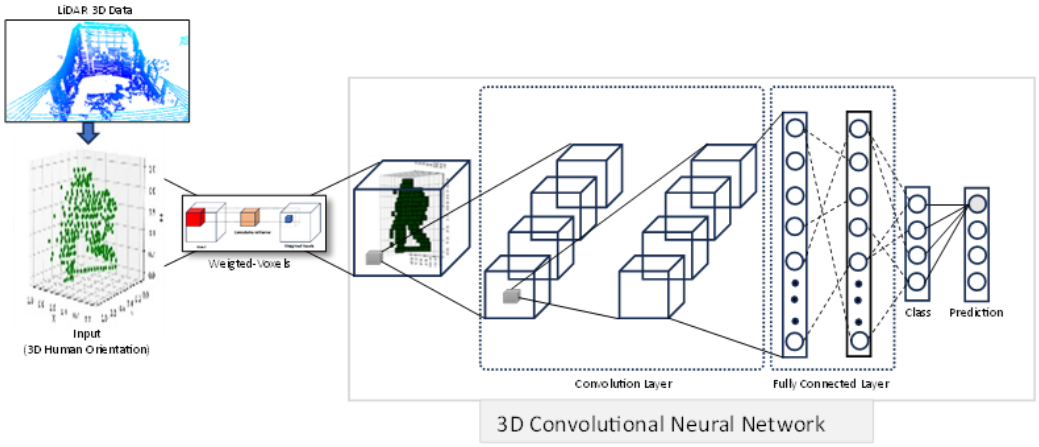


Figure 5. The proposed architecture

This research aims to apply Convolutional Neural Network (CNN) architecture in classifying human orientation based on 3D Point Cloud data. Deep learning research using Point Cloud data has involved various architectures with the goal of achieving reliable results. The main proposed architecture includes the addition of weighted voxel conversion at the

beginning before entering several CNN architectures. In this study, we will test five different CNN architectures with both the public KITTI dataset and our dataset. These datasets have undergone feature extraction from raw Point Cloud data and will be used for architecture performance comparisons. The five CNN architectures to be evaluated are VGG16, Resnet50, Resnet50v2, DenseNet121, and VoxNet, aiming to obtain the best architecture for orientation estimation with weighted voxel combination.

Table 1. Layer Modifications in Each Evaluated Architecture

Modified	**VGG16**	**ResNet50**	**ResNet50v2**	**DenseNet121**	**VoxNet**
Input	(224,224,3) ↓ (16,16,16,1)	(224,224,3) ↓ (16,16,16,1)	(224,224,3) ↓ (16,16,16,1)	(224,224,3) ↓ (16,16,16,1)	**Not Modified**
Convolution	Conv2D ↓ Conv3D	Conv2D ↓ Conv3D	Conv2D ↓ Conv3D	Conv2D ↓ Conv3D	
Maxpooling	Maxpooling 2D ↓ Maxpooling 3D	Maxpooling 2D ↓ Maxpooling 3D	Maxpooling 2D ↓ Maxpooling 3D	Maxpooling 2D ↓ Maxpooling 3D	
ZerroPadding	-	ZerroPadding 2D ↓ ZerroPadding 3D	ZerroPadding 2D ↓ ZerroPadding 3D	ZerroPadding 2D ↓ ZerroPadding 3D	
Depthwise Convolution	-	DepthwiseConv 2D ↓ DepthwiseConv 3D	DepthwiseConv 2D ↓ DepthwiseConv 3D	-	

VGG16 is one of the architectures developed that is consists of 13 convolution layers, with max-pooling layers interspersed between each convolution layer. After feature extraction, the data enters three fully connected neural network layers and ends with a softmax layer for classification. Resnet is known as a deep network that leverages the concept of residual layers to improve classification accuracy. ResnetV2, a variant of Resnet, combines residual layers with batch normalization to enhance performance compared to the previous Resnet. On the other hand, Desnet is a deep network with 56 layers that use padding on each layer and utilize instance normalization as an alternative to batch normalization. Desnet implementation has been proven to significantly improve image classification.

All architectures have been modified into 3D CNNs to process input data in voxel format in three-dimensional space. The convolution layers, which are the core elements of each architecture, have been adapted into three-dimensional kernel layers to handle the characteristics of 3D point cloud data.

5. EXPERIMENT AND ANALYSIS

In this section, we will explain the dataset base used in the testing for 3D object classification, particularly human orientation. In the following explanation, we will present the detailed implementation of the proposed method, which includes system parameters set for testing. Furthermore, we will discuss the experimental results compared to previous approaches. It is essential to note that the results being compared are as reported by the authors in the respective papers.

5.1 Datasets

In this experiment, we employed two dataset options: a public dataset and our self-created primary dataset. The public dataset used is the KITTI dataset[24] for 3D LiDAR data of human individuals. This public dataset is often utilized to compare the performance of various architectures for various 3D object functions. Additionally, we utilized our self-collected primary dataset using LiDAR sensors to classify human orientation into four categories.

Table 2. The number of public and our datasets used in the experiments

Dataset	Class	Total	Training	Testing
KITTI Dataset	North	40	31	8
	South	54	43	11
	West	68	56	12
	East	300	66	16
Our Dataset	North	365	250	115
	South	250	150	100
	West	270	200	70
	East	300	250	50

We used a manual cropping method to obtain human data and clean other data in the surroundings captured by the LiDAR sensor, both in the public dataset and our dataset. The number of datasets from each source is presented in Table 2, and Figure 6 illustrates how the dataset is categorized into four orientation classes: North, South, West, and East, based on the 3D LiDAR positions, as explained in the previous introduction section.

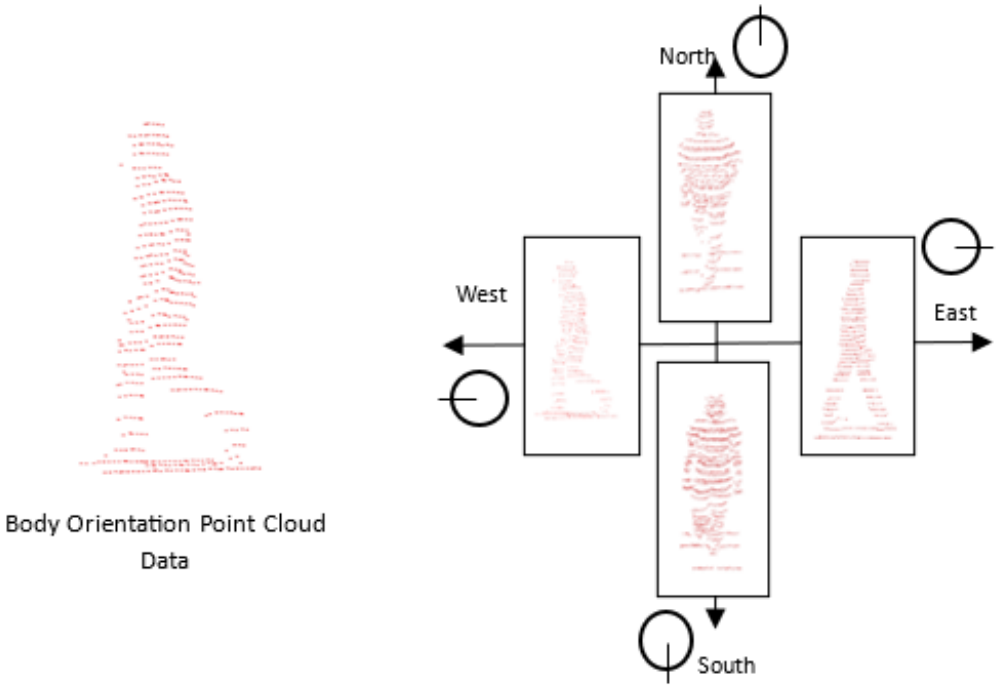


Figure 6. Categories of 3D Point Cloud Human Orientation

5.2 Implementation Details

Our experiments were conducted using the Ouster LiDAR sensor with 32 channels for data collection, and we utilized the NVIDIA Tesla T4 GPU available on the Kaggle cloud platform for the learning process, which provides free but limited access. We implemented the neural network using the TensorFlow-gpu API V2.11.0, along with several supporting libraries in the Python platform. We applied the one-cycle learning rate policy during the training process, with the training cycle set to 50 epochs. Additionally, we configured the learning rate to be 0.0001 and the batch size to be 10. We also employed 5-fold cross-validation. Due to the dataset's class imbalance, we used the Synthetic Minority Over-sampling Technique (SMOTE), a popular method for balancing data. This technique synthesizes new samples from the minority class to balance the dataset by creating new items from the minority class with the formation of convex combinations of nearby items.

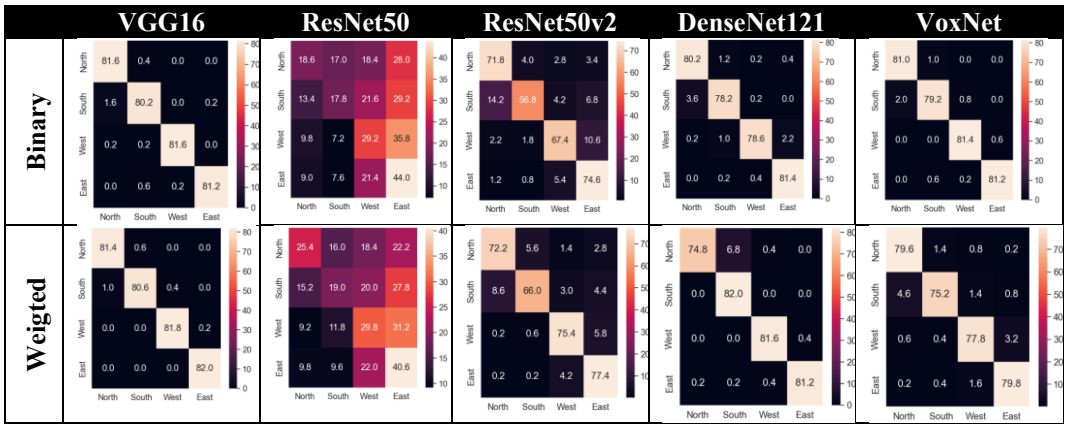
5.3 3D Human Orientation Classification

In this study, we first evaluated human orientation classification using several CNN architectures: VGG16, ResNet50, ResNet50v2, DenseNet121, and VoxNet[25]. We employed two types of point cloud conversion into voxel format: the first type with binary voxels, where filled voxels are assigned a value of 1 and empty voxels a value of 0. The second type is weighted voxels, where we used values ranging not only between 0 and 1 but also specific integer values, including negatives and positives, to provide integer values to the voxels. This testing represents an extension of previous research that

involved only four CNN architectures with binary voxels. In the previous research, VGG16 outperformed the other three architectures in terms of accuracy and loss. Furthermore, this time, we added a comparison with another new architecture, VoxNet, which is a voxel-based architecture often used for comparisons.

The purpose of using weighted voxels is to enhance the performance of CNN architectures, especially for human orientation estimation using point cloud data. We present a comparison between binary voxels and weighted voxels for the specified architectures in the form of confusion matrices for testing the classification of four orientation classes. The classification tests were conducted by comparing two datasets to determine how robust the architectures are to different data, significantly when their performance is enhanced with the addition of weighted voxels. Based on Table 3 and Table 4, the approach of adding weighted voxels to each architecture outperforms the use of binary voxels in CNN architectures. The confusion matrix results show that the top two, VGG16 and DenseNet12, have high average classification success rates for each class. This is evident from the dominant colors forming diagonal patterns, indicating better accuracy compared to the other three architectures. Interestingly, there is a different outcome for the VoxNet architecture, which experiences a decrease in classification performance when using weighted voxels. However, an overall observation shows that the use of weighted voxels actually improves orientation classification performance in every architecture except for VoxNet.

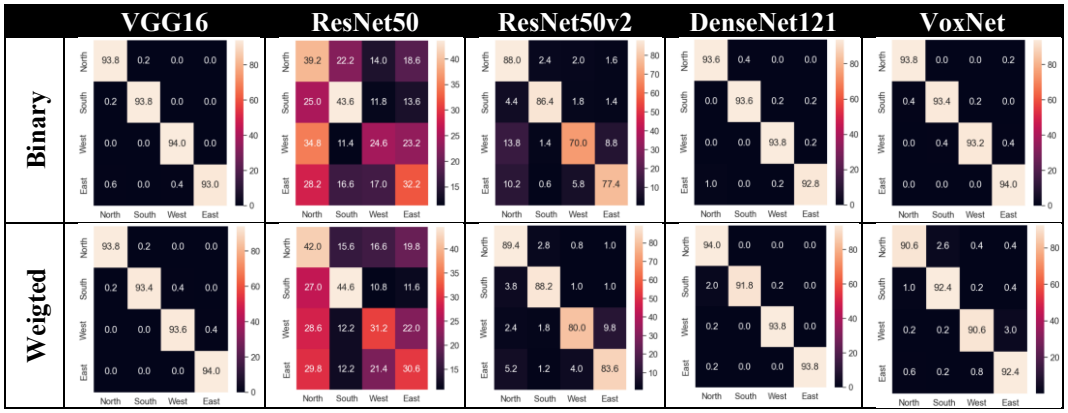
Table 3. Comparison of results between binary voxels and weighted voxels combined with CNN architectures using confusion matrices on the public KITTI dataset



Based on training data from the KITTI dataset for each class, including North, South, West, and East, we observed that ResNet50 has a very low success rate for each class compared to the other architectures. For the binary voxels, the success rates are as follows: North 18.6%, South 17.8%, West 29.2%, and East 44%. Similarly, with weighted voxels, although there is an

improvement in three classes, the success rates remain low North 25.4%, South 19%, and West 29.8%. In contrast, ResNet50v2 shows an average increase of 5.1% in success rates for each class using the KITTI training data. For the VGG16, DenseNet121, and VoxNet architectures, there are difficulties with the North class, which actually experiences a decrease in success rate with the addition of weighted voxels. However, the other classes show improvement, which positively affects the overall classification success rate. Furthermore, when comparing these results with our own collected data, the success rates for each class in each architecture exhibit similar characteristics to those observed with the public dataset.

Table 4. Comparison of results between binary voxels and weighted voxels combined with CNN architectures using confusion matrices on our self-acquired dataset from the 3D LiDAR sensor



Upon evaluating all architectures, it is clear that ResNet exhibited the lowest classification results for each class in this particular scenario. We also provide performance results in terms of accuracy and loss, based on the evaluation during training, using two different datasets: the KITTI dataset and our dataset. Upon reviewing Tables 5 and 6, a noticeable trend is the significant improvement in accuracy and reduction in loss for each architecture. However, in the case of the VoxNet architecture, the results differ with the addition of weighting, which, surprisingly, can negatively impact performance. The VoxNet architecture is considerably more straightforward than the other architectures and has fewer parameters, which might result in faster training computations compared to the other architectures we assessed. The incorporation of structural information through weighted voxels significantly enhances the performance of architectures with a larger number of parameters. This is particularly evident in the case of VoxNet, where the results suggest that simpler architectures benefit more from using binary voxels due to their straightforward and easily interpretable data structure.

Table 5. The Classification Results from KITTI dataset

Training Dataset	Model Type	Preprocessing	Accuracy (%)	Loss (%)
KITTI Dataset	ResNet50	Binary Voxel	31.39	1.69
		Weighted Voxel	**32.59**	**1.57**
	ResNet50V2	Binary Voxel	50	1.36
		Weighted Voxel	**67.69**	**0.9**
	DenseNet121	Binary Voxel	90.26	0.39
		Weighted Voxel	**95.14**	**0.18**
	VoxNet	Binary Voxel	93.6	0.2
		Weighted Voxel	**80.78**	**0.52**
	VGG16	Binary Voxel	96.04	0.35
		Weighted Voxel	**98.17**	**0.15**

Table 6. The Classification Results from our dataset

Training Dataset	Model Type	Preprocessing	Accuracy (%)	Loss (%)
Our Dataset	ResNet50	Binary Voxel	32.72	1.4
		Weighted Voxel	**34.55**	**1.38**
	ResNet50V2	Binary Voxel	58.23	1.16
		Weighted Voxel	**71.29**	**0.68**
	DenseNet121	Binary Voxel	97.6	0.1
		Weighted Voxel	**98.4**	**0.06**
	VoxNet	Binary Voxel	97.87	0.09
		Weighted Voxel	**88.55**	**0.32**
	VGG16	Binary Voxel	98.41	0.09
		Weighted Voxel	**98.67**	**0.07**

Overall, it can be observed that the combination of weighting with the VGG16 CNN architecture experienced a significant performance increase, reaching an accuracy of 98.67% compared to other architectures. The results of DenseNet121 can also be considered competitive with VGG16 CNN, as it achieved an accuracy of 98.4% the second-highest. Meanwhile, VoxNet experienced a decrease in accuracy with the presence of weighted voxels, dropping from its original 97.87% to 88.55%. These results suggest that the combination of VGG16 with weighted voxel preprocessing can be relied upon for estimating human orientation based on 3D point cloud data.

Table 7. The Computation time of each model

Model Type	**Computation Time (s)**
VGG16	762.8
ResNet50	262.5
ResNet50v2	262.5
DenseNet121	864.6
VoxNet	58.2

We also provide a comparison of the training computation time for each model to offer a promising perspective on the effectiveness of orientation classification. As seen in Table 7, DenseNet121 has the longest computation time compared to the other models. The shortest computation time is observed in VoxNet, with a recorded time of 58.2 for a single training cycle. This computation speed is influenced by the number of parameters in each architecture. The modified VGG16 demonstrates a mid-range computation speed but with better performance compared to the other models.

6. CONCLUSION

In this paper, we proposed the addition of preprocessing, namely weighted voxels, to several CNN architectures used for estimating human orientation based on 3D point cloud data. A CNN-based approach was combined with weighted voxels for the task of capturing and classifying 3D human orientation. We employed weighted voxels to transform information from 3D binary voxel data into voxels with robust and discriminative descriptors. The performance of weighted voxels was reinforced by comparing their performance using convolution matrices on the KITTI public dataset and our dataset. We divided this dataset into four orientation classes. The comparison between the use of weighted voxels and binary voxels in several architectures, such as VGG16, ResNet50, ResNet50V2, DenseNet121, and VoxNet achieved competitive performance in a series of experiments. In our ongoing research, we will continue to explore voxel-based preprocessing approaches with specific feature modifications, not limited to binary or weighted, and also develop other deep learning architectures.

Acknowledgments

This work has been fully funded and supported by Balai Pembiayaan Pendidikan Tinggi (BPPT) under the Ministry of Education, Culture, Research, and Technology, as well as Lembaga Pengelola Dana Pendidikan Indonesia and in part of the Penelitian Fundamental - Riset Dasar Research Grant.

REFERENCES

[1] Banerjee A, Galassi F, Zacur E, De Maria GL, Choudhury RP, and Grau V, **Point-Cloud Method for Automated 3D Coronary Tree**

Reconstruction From Multiple Non-Simultaneous Angiographic Projections, *IEEE Trans Med Imaging*, vol. 39, pp. 1278–90, 2020.

[2] Han L, Zheng T, Zhu Y, Xu L, and Fang L, **Live Semantic 3D Perception for Immersive Augmented Reality**, *IEEE Trans Vis Comput Graph,* vol. 26, pp. 2012-2022, 2020.

[3] Li J, Qin H, Wang J, and Li J, **OpenStreetMap-Based Autonomous Navigation for the Four Wheel-Legged Robot Via 3D-Lidar and CCD Camera**. *IEEE Transactions on Industrial Electronics*, vol. 69, pp. 2708-2717, 2022.

[4] Zeng Y, Hu Y, Liu S, Ye J, Han Y, Li X, Sun N, **RT3D: Real-Time 3-D Vehicle Detection in LiDAR Point Cloud for Autonomous Driving**, *IEEE Robot Autom Lett*, vol. 3, pp. 3434-3440, 2018.

[5] Ma L, Li Y, Li J, Yu Y, Junior JM, Goncalves WN, Chapman MA., **Capsule-Based Networks for Road Marking Extraction and Classification From Mobile LiDAR Point Clouds**, *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, pp. 1981-1995, 2021.

[6] Caesar H, Bankiti V, Lang AH, Vora S, Liong VE, Xu Q, Krishnan A, Pan Y, Baldan G, Beijbom O, **nuScenes: A multimodal dataset for autonomous driving**, 2019.

[7] Duan Y, Zheng Y, Lu J, Zhou J, and Tian Q, **Structural Relational Reasoning of Point Clouds**, *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 949–58, 2019.

[8] Song X, Wang P, Zhou D, Zhu R, Guan C, Dai Y, Su H, Li H, Yang R, **ApolloCar3D: A Large 3D Car Instance Understanding Benchmark for Autonomous Driving**, 2018.

[9] Lv C, Lin W, and Zhao B, **Voxel Structure-Based Mesh Reconstruction From a 3D Point Cloud**, *IEEE Trans Multimedia*, vol. 24, pp. 1815-1829, 2022.

[10] Kang Z, Yang J, Zhong R, Wu Y, Shi Z, and Lindenbergh R, **Voxel-Based Extraction and Classification of 3-D Pole-Like Objects From Mobile LiDAR Point Cloud Data**, *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 11, pp. 4287-4298, 2018.

[11] Agrawal S, Bhanderi S, Doycheva K, and Elger G. **Static Multitarget-Based Autocalibration of RGB Cameras, 3-D Radar, and 3-D Lidar Sensors**, *IEEE Sens J,* vol. 23, pp. 21493-21505.

[12] Kettelgerdes M, and Elger G, **In-Field Measurement and Methodology for Modeling and Validation of Precipitation Effects on Solid-State LiDAR Sensors**, *IEEE Journal of Radio Frequency Identification*, vol. 7, pp. 192-202, 2023.

[13] Liu W, Tang X, and Zhao C, **Robust RGBD Tracking via Weighted Convolution Operators**, *IEEE Sens J*, vol. 20, pp. 4496-4503, 2020.

[14] Sun W, Iwata S, Tanaka Y, and Sakamoto T, **Radar-Based Estimation of Human Body Orientation Using Respiratory Features and Hierarchical Regression Model**, *IEEE Sens Lett*, vol. 7, pp. 1-4, 2023.

[15] Cardarelli S *et al*, **Single IMU Displacement and Orientation Estimation of Human Center of Mass: A Magnetometer-Free Approach**, *IEEE Trans Instrum Meas*, vol. 69, pp. 5629-5639, 2020.
[16] Li S, Li L, Shi D, Zou W, Duan P, and Shi L, **Multi-Kernel Maximum Correntropy Kalman Filter for Orientation Estimation**, *IEEE Robot Autom Lett*, vol. 7, pp. 6693-6700, 2022.
[17] Zhang J-H, Li P, Jin C-C, Zhang W-A, and Liu S, **A Novel Adaptive Kalman Filtering Approach to Human Motion Tracking With Magnetic-Inertial Sensors**, *IEEE Transactions on Industrial Electronics*, vol. 67, pp. 8659-8669, 2020.
[18] Fisch M and Clark R, **Orientation Keypoints for 6D Human Pose Estimation**, *IEEE Trans Pattern Anal Mach Intell*, vol. 44, pp. 10145-10148, 2022.
[19] Lee D, Yang M-H, and Oh S, **Head and Body Orientation Estimation Using Convolutional Random Projection Forests**, *IEEE Trans Pattern Anal Mach Intell*, vol. 41, pp. 107-120, 2019.
[20] Wu C, Chen Y, Luo J, Su C-C, Dawane A, Hanzra B, Deng Z, Liu B, Wang J, Kuo C-H, **MEBOW: Monocular Estimation of Body Orientation In the Wild**, 2020.
[21] Riansyah MochI, Sardjono TA, Yuniarno EM, and Purnomo MH, **Prediction of Human Body Orientation based on Voxel Using 3D Convolutional Neural Network**, *2023 International Seminar on Intelligent Technology and Its Applications (ISITIA), IEEE*, pp. 99–104 2023.
[22] Xie H, Yao H, Sun X, Zhou S, and Tong X, **Weighted voxel**, *Proceedings of the 10th International Conference on Internet Multimedia Computing and Service*, New York, pp. 1–4 2018.
[23] Dewantara BSB, Saputra RWA, and Pramadihanto D, **Estimating human body orientation from image depth data and its implementation**, *Mach Vis Appl*, vol. 33, pp. 38, 2022.
[24] Menze M, and Geiger A, **Object scene flow for autonomous vehicles. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, *IEEE*, pp. 3061–70, 2015.
[25] Maturana D and Scherer S, **VoxNet: A 3D Convolutional Neural Network for real-time object recognition**, *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE*, pp. 922–928, 2015.