IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 17 April 2024, accepted 13 May 2024, date of publication 17 May 2024, date of current version 30 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3402227

RESEARCH ARTICLE

Utilizing a Single-Stage 2D Detector in 3D LiDAR Point Cloud With Vertical Cylindrical Coordinate Projection for Human Identification

NOVA EKA BUDIYANTA[1,2], (Student Member, IEEE), EKO MULYANTO YUNIARNO[1,3], (Member, IEEE), TSUYOSHI USAGAWA[4], (Member, IEEE), AND MAURIDHI HERY PURNOMO[1,3,5], (Senior Member, IEEE)

[1]Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia
[2]Department of Electrical Engineering, Universitas Katolik Indonesia Atma Jaya, Jakarta 12930, Indonesia
[3]Department of Computer Engineering, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia
[4]Graduate School of Science and Technology, Kumamoto University, Kumamoto 860-8555, Japan
[5]University Center of Excellence on Artificial Intelligence for Healthcare and Society (UCE AIHeS), Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia

Corresponding author: Mauridhi Hery Purnomo (hery@ee.its.ac.id)

This work was supported in part by BPI (BPPT)-LPDP Scholarship, and in part by Institut Teknologi Sepuluh Nopember in collaboration with Kumamoto University.

This work involved human subjects or animals in its research. The authors confirm that all human/animal subject research procedures and protocols are exempt from review board approval.

ABSTRACT Exploiting sensitive human data in human visual monitoring system violates individual's privacy. Hence, this study utilized a Light Detection and Ranging (LiDAR)-generated three-dimensional (3D) point cloud instead of an RGB camera as it captures the human object without detailed imagery. Given their dispersed nature, processing 3D LiDAR point cloud is economically inefficient as it requires some preliminary actions. Alternatively, this study applied a single-stage detection process using only one data type, namely 3D LiDAR point cloud with vertical axis direct projection. This approach utilized the 3D LiDAR cylindrical coordinates features to project the data onto two-dimensional (2D) image space with various computational capabilities and back-projection algorithm to restore the identified object on the 2D plane onto 3D LiDAR point cloud coordinates. This model also implemented the YOLOv5 series due to their varied sizes. The evaluation of this approach utilized accuracy metric which was the average of mean average precision (*mAP*) value based on different intersections over union (*IoU*) thresholds. The proposed methodology effectively employed a vertical projection technique to identify human objects. Notably, this approach distinguishes itself from previous methods such as PIXOR, BirdNet, BirdNet+, BEVDetNet, and Frustum-PointPillars, offering a novel perspective in the field. In addition, the best and worst performing models had the accuracy values of 44.35% and 79.83%, with inference speeds of 3.7 ms and up to 25 ms, respectively. Further, the inference speeds of all models were less than 33.33 ms. Thus, the monitored objects were identified before the LiDAR system enters the next azimuth rotation.

INDEX TERMS 2D projection, 3D LiDAR point cloud, computation capabilities, cylindrical coordinates, human identifier.

I. INTRODUCTION

Human identification process is a fundamental concern in several fields, including in autonomous driving safety systems, anomaly human movement in urban areas, 3D multimedia, collaborative robots, and medical rehabilitation. In autonomous driving safety systems, human identification helps to reduce accidents through vision-based crossing analysis [1] and gait analysis [2]. In urban areas, it detects anomaly human movement that plays a vital role in theft

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

detector [3]. For 3D multimedia, it contributes to the creation of 3D animation character motion [4]. Additionally, in collaborative robots, it facilitates the interaction between robots and humans [5]. Moreover, in medical rehabilitation, it supports fall detection [6] and monitors patient activities [7], [8].

In many of these applications, particularly in human activity monitoring, the process often relies on 2D RGB camera images [9]. Unfortunately, this approach has limitations, primarily due to privacy issues [10]. Using an RGB cameras can be intrusive as it captures detailed physical appearances and the surroundings [11]. This may discomfort the human object as it records and potentially exposes personal and private moments. Furthermore, RGB cameras captures clear images so that this approach lacks of anonymity. This can be a concern where individuals expect privacy, such as in a restroom or changing room.

Using cameras also potentially impacts the accuracy of the data. To gather depth information in images, researchers often use photogrammetry which involves several cameras and projects the data from various angles [12], [13], [14]. However, this method often faces several accuracy-related issues. First, its accuracy highly depends on the quality of the images, which are affected by several conditions such as lighting, camera resolution, and lens distortion. Second, occlusions in dynamic environments such as in human monitoring system can lead to incomplete data capture and errors in measurement. The images captured from different angles can also result on perspective distortions. In other words, employing photogrammetry encounters challenges particularly when the object being monitored is human.

Alternatively, Time-of-Flight (ToF) scanner sensors, including Light Detection and Ranging (LiDAR), are increasingly used for human activity monitoring [15], [16]. Unlike RGB cameras, LiDAR provides measures distances using laser light which allows more accurate tracking and analysis of human movements and positions. In addition, it is able to preserve the individual's privacy while being monitored as it does not capture detailed visual features. A LiDAR creates point clouds representing the shape and size of objects, including people, without revealing identifiable features like faces. This lack of visual detail makes it less intrusive and helps preserve the anonymity of individuals being monitored.

LiDAR is available in both two-dimensional (2D) and three-dimensional (3D) versions [17], [18]. The 2D LiDAR generates a point cloud consisting of x and y coordinates, which represent the locations of points on a 2D plane [19]. On the other hand, 3D LiDAR incorporates a z-coordinate, enabling the representation of points in 3D space, hence providing a more comprehensive depiction of the surroundings [20]. Each point within these cloud formations represents a consistent distance, providing a more dependable measurement in contrast to the approximations employed in camera-based photogrammetry approaches [21]. Furthermore, it adeptly records human movements throughout a period, offering a precise depiction of gestures [22]. Hence, the utilization of 3D LiDAR holds great importance.

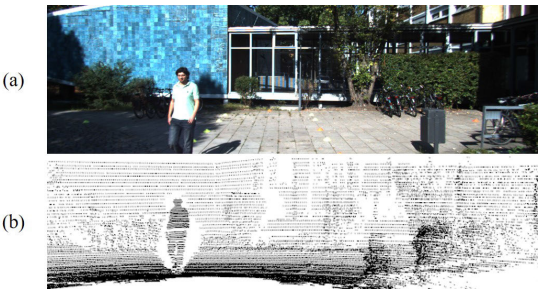

FIGURE 1. Comparison between environment captured in: (a) Camera pixel; (b) 3D LiDAR point cloud [24].

Both 2D and 3D LiDAR systems utilize lasers that are emitted and subsequently reflected back to a receiver, providing a complete 360° horizontal field of view (HFoV) [23]. The primary distinctions between 2D and 3D LiDAR systems are found in their laser channels, receivers, and fields of view (FoVs). A 2D LiDAR system consists of a laser and receiver that spin around an axis. This setup typically offers a single horizontal field of view (HFoV) channel, enabling the device to scan the surrounding environment in a flat, two-dimensional plane. Conversely, 3D LiDAR systems possess several laser and receiver channels, allowing them to collect data over a vertical field of view (VFoV) as well. Hence, 3D LiDAR has the capability to scan objects in both horizontal and vertical directions, resulting in the generation of a comprehensive three-dimensional representation of the environment. Figure 1 visualizes the environment captured by Camera and LiDAR.

Despite its advantages, utilizing 3D LiDAR point clouds presents processing challenges due to the dispersed nature. The point clouds frequently exhibit a dispersed arrangement due to the diverse placements and complexities they capture [25]. The spatial distribution and luminosity of points inside the cloud can fluctuate depending on the object's dimensions, surface reflectance, and proximity to the LiDAR sensor [26]. Moreover, the processing becomes challenging when the point cloud is insufficient, which may occur due to object obstruction or a cluttered background leading to occlusions [27]. In addition, accurately interpreting each component in the cloud can be challenging, especially when dealing with similar surfaces or forms, or when objects have equal reflective properties.

Various methods have been extensively investigated to identify human activity in surveillance systems with 3D LiDAR. Nevertheless, these methodologies frequently face computational obstacles such as the substantial computational expense [28]. This is associated with the challenges in managing point cloud data and utilizing deep learning models, which can impose a significant load [29].

A. MOTIVATION

Dealing with point cloud data presents significant difficulties because of the existence of permutation and orientation

invariants [30], as well as rigid transformations [31], [32], such as 3D transformations and rotations. Furthermore, deep learning algorithms that employ LiDAR point cloud data may encounter challenges in managing substantial volumes of data. The issue of accuracy and data quality in object detection using 3D LiDAR must be acknowledged and resolved [33]. Additionally, it is important to recognize that the efficiency could be diminished as a result of limited processing capabilities and storage capacity [34]. A commonly used approach to overcome these challenges is using bird's eye view (BEV) geometry projection, which provides a horizontal perspective from an elevated position. Various approaches have been developed based on this method, including the PIXOR [35], BirdNet [36], BirdNet+ [37], [38], BEVDetNet [39], and Frustum-PointPillars approach [40].

The PIXOR approach used 3D LiDAR Point Cloud from the KITTI dataset to be horizontally projected and processed to enable object detection using single-stage 2D detector. The BirdNet approach, meanwhile, using 3D point cloud data from the LiDAR sensor and points representing the ground as the input to be projected to Bird's Eye View (BEV) image. This approach is then developed as BirdNet+ which uses two features from BEV projection to encode the height and occupancy. The maximum heights of the 3D points are then mapped to pixels that suit the BEV image. The value of an occupancy plane is 0 if no 3D points are mapped to the plane. Otherwise, the value is 1. This channel enables the network to explicitly cover invalid areas. Similarly, the BEVDetNet is the development of the BirdNet+ which involves height and occupancy generated from the projection of 3D LiDAR point cloud BEV. However, BEVDetNet employs intensity canal to be processed in 2D single-stage detector so that 3D object center keypoints can be gained. The latest development of these approaches is called Frustum-PointPillars which uses two types of data, namely projected 3D LiDAR point cloud BEV and RGB image. In this approach, double-stage 2D detector is implemented for RGB image while the projected BEV is used to reconstruct vertical pillar based on the frustum resulted from 2D bounding box of the RGB image to detect object. These methods involve using voxelized point cloud data to create binary bird's-eye-view (BEV) maps [41], [42].

The BEV technique improves data accessibility by structuring the point cloud data, allowing for effective problem-solving in path planning, collision avoidance, and precise position estimates of car objects [43]. Unfortunately, it is inadequate to accurately see the entirety of human body motions to track human activities. For instance, in the pedestrian detection, BEV implementation might be challenged with limited vertical data. Moreover, as the result of this method is a bounding box that may expose unnecessary surrounding points, it can lead to incomplete or inaccurate data representation [37], [38].

As an alternative, the Frustum-PointPillars technique [40] can be used as it offers vertical information on the object under surveillance. Nevertheless, an RGB camera is necessary for this approach since it integrates the approaches of Frustum PointNets and PointPillars, which depend on distinct data inputs and processing techniques. As a result, employing this method leads to higher expenses in computer processing. Table 1 accurately depicts the specific details of the comparison of techniques. Considering the problems that have not been successfully addressed, this study aims to provide an alternative that solve the issues.

B. MAIN CONTRIBUTION

This study presents a novel approach to improve the utilization of 3D LiDAR point cloud data for human identification in monitoring systems. The primary objective is to devise a methodology that achieves a harmonious equilibrium between the precision of the model and the speed at which it operates, while accommodating different levels of computational capabilities. In many cases when the device promotes high accuracy, the speed is very low because it requires longer time to proceed huge amount of data. Meanwhile, in this study, the speed is improved through the implementation of single-stage 2D detector while maintining the most optimum accuracy. Therefore, the proposed method can minimize the data processing expenses and efficiently manage the scattered characteristics of 3D LiDAR point cloud data, particularly when employing 2D detectors. The 3D LiDAR-generated point cloud retains its integrity when transformed into 2D space due to its cylindrical cross-sections [44]. In other words, the novelty of this proposed approach lies in the utilization of one data type, namely 3D LiDAR point could that is implemented in single-stage 2D detector as an alternative to improve the efficiency of human detection without invading the objects' privacy.

Furthermore, this study aims to ascertain the optimal equilibrium between velocity and precision, and to create an efficient model that can be utilized for future monitoring tasks using 3D LiDAR point cloud data. Hence, this research makes significant contributions to the science of human identification in multiple crucial aspects:

- Creating a procedure to preprocess raw 3D point cloud data for identification by a single-stage 2D detector.
- Investigating the capabilities of perspective projection techniques in the human identification.
- Transforming dispersed 3D LiDAR point cloud data into a format that is appropriate for a 2D detector model.
- Reconstructing the 3D LiDAR point cloud by mapping the detected 2D human figurines onto it.
- Performing reliability assessments on the detection technique using various dimensions of 2D detectors to evaluate their speed and precision.

The paper continues as follows: Section II discusses related works that support our contribution; Section III describes our proposed methods for object detection on 3D to 2D vertical axis projection; Section IV presents the results of the

TABLE 1. 3D LiDAR point cloud object identifier methodology comparison.

Methods	Dataset	Data Type	Projection Type	2D Detector Type	Object(s)
PIXOR [35]	KITTI	3D LiDAR Point Cloud	Horizontal (BEV)	Single-stage	Car
BirdNet [36]	KITTI	3D LiDAR Point Cloud	Horizontal (BEV)	Double-stage	Car, Human, Cyclist
BirdNet+ [37]	KITTI	3D LiDAR Point Cloud	Horizontal (BEV)	Double-stage	Car, Human, Cyclist
BEVDetNet [39]	KITTI	3D LiDAR Point Cloud	Horizontal (BEV)	Single-stage	Car
Frustum-PointPillars [40]	KITTI	3D LiDAR Point Cloud + RGB Image	Horizontal (BEV) for 3D LiDAR Point Cloud + Vertical for RGB Image	Double-stage	Car, Human, Cyclist
Proposed	KITTI	**3D LiDAR Point Cloud**	**Vertical**	**Single-stage**	**Human**

experimental phase; and Section V concludes with a summary and application considerations.

II. RELATED WORK

A. 3D LIDAR POINT CLOUD DATA APPLICATIONS

3D LiDAR, extensively employed in diverse research domains, produces significant point cloud data. The KITTI dataset [24] serves as an illustration, which was acquired via a 3D LiDAR Velodyne HDL-64E. This specific LiDAR model conducts a 360° HFoV every 100 ms, resulting in a sample rate of 10 times per second. In addition, it provides comprehensive spatiotemporal data using its discrete 3D scanning feature.

The spatial characteristics of the data are determined by two essential parameters: vertical and azimuthal resolutions. The vertical resolution is obtained by utilizing 64 laser beams, evenly distributed across a VFoV spanning 26.8°. The laser beams are spaced apart by a vertical angle of 0.4°. The azimuth resolution is attained by conducting 360° horizontal scans, with each scan taking place at intervals of 0.09°. The LiDAR gathers 64 rows and 4000 columns of distance data every scan from its 64 channels. The extensive data serves as the foundation for the coordinates of the point cloud.

A 3D point cloud is generated using LiDAR technology and is then mapped onto a cylindrical surface with a hollow interior [45]. This projection method guarantees that the LiDAR lasers avoid scanning the unexposed side of an object, resulting in areas where data is absent. Figure 2 depicts the technical configuration of a 3D LiDAR system in a visual format.

The LiDAR data is analyzed in terms of both its spatial and temporal dimensions in order to create a point cloud that is derived from the recorded distances of reflection. The points can be represented using several coordinate systems, such as cartesian coordinates (x, y, z) [46], [47], cylindrical coordinates (r, θ, z) [48], or spherical coordinates (θ, ϕ, r) [47]. In cylindrical and spherical coordinates, θ denotes the azimuth angle in the x,y plane. However, in spherical coordinates, θ also represents the inclination angle along the z-axis in 3D space. Moreover, the r value in these coordinate systems is crucial since it denotes the distance between the surface reflection point and the LiDAR sensor.

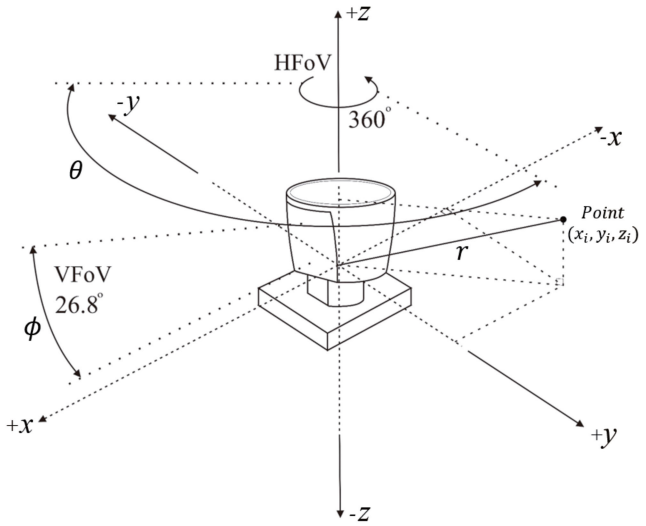

FIGURE 2. The technical architecture of a 3D LiDAR system that encompasses the laser emission and reflection towards objects within a horizontal field of view (HFoV) of 360° and a vertical field of view (VFoV) of 26.8°.

The 3D point cloud data obtained from a LiDAR system comprises laser reflections, where each reflection is recorded as an individual point in a dispersed arrangement. In the KITTI dataset [24], the 3D LiDAR point cloud data is represented by 4-tuple elements. These elements consist of cartesian coordinates and a p intensity value (x, y, z, p). The intensity value, denoted as p, represents the surface properties of objects that are impacted by the LiDAR laser beam. In order to utilize this data optimally, it is necessary to normalize the p values, thereby changing them to a scale ranging from 0 to 1.

B. PREVIOUS 3D POINT CLOUD PROCESSING

Numerous studies have investigated many approaches to tackle the challenges in processing point cloud data, such as through point cloud segmentation [49] and motion detection [50]. Due to the data's dispersed nature, it requires preprocessing before the utilization in computational models. Clustering is a widely used technique for processing point clouds. One often used algorithm for clustering is k-means,

which uses basic metrics like Euclidean distance to group points and assign labels [51].

The processing of point cloud data is constantly advancing. Initially developed for image data, Convolutional Neural Networks (CNNs) necessitate organized input. As a result, researchers should preprocess point cloud data by employing voxelization [52], [53], [54]. Nevertheless, this preprocessing can diminish the computing efficiency.

Several techniques have been devised that enable the direct manipulation of point cloud data without voxelization [55], [56]. An exemplary instance is PointNet [56], which has demonstrated efficacy in diverse point cloud processing applications, such as 3D object recognition and segmentation. However, it is inefficient as it requires an extensive preprocessing and more computational resources. This paper presents a novel method for identifying human things in 3D point cloud data using 2D detectors. This strategy is designed to overcome the limitations or obstacles.

C. 2D OBJECT DETECTORS

Several object detectors using CNN have been developed to detect and identify objects in 2D images. These detectors utilize a double-stage methodology, such as Region-based CNNs (R-CNNs) [57], which initially employed selective search [58] to propose regions. Although this method is effective, it is time-consuming. To address the problem, Fast R-CNN [59] was introduced to improve speed, achieving processing times that are 25 times faster than R-CNN. Subsequently, Faster R-CNN [60] was introduced, boasting a remarkable speed improvement of 250 times compared to R-CNN. This achievement is attributed to its capability of generating region proposals internally, hence eliminating the requirement for an external proposer. Nevertheless, Faster R-CNN's efficiency can be hindered by the extensive processing of several areas suggested by its Region Proposal Network (RPN). In order to enhance speed while maintaining accuracy, researchers developed single-stage detectors such as You Only Look Once (YOLO) [61] and Single Shot Detector (SSD) [62]. These techniques employ a collaborative grid-based strategy for rapid identification.

Although these approaches are fast, they encounter difficulties in precisely forecasting bounding boxes for small clusters of objects [61]. In order to tackle this issue, the SSD method underwent improvements by using multiscale representations [63], multiple anchor boxes [64], and the RPN from Faster R-CNN [60]. This entails distinguishing bounding box outputs according to diverse aspect ratios and predetermined anchor scales, as well as combining several feature maps with differing resolutions to enhance the detection of objects with different sizes. Nevertheless, this method may have difficulties in achieving precise results, particularly when dealing with tiny things, and also has efficiency challenges when handling bigger objects. Additionally, it has a tendency to disregard characteristics at the fourth convolution level, which has a negative impact on the total accuracy of predictions.

Thus far, YOLO has mainly been used to identify 2D objects. Compared to the other lightweight models mentioned above, the YOLO software versions are more compatible either with high or low hardware computing levels. By implementing EfficientDet [65] to adjust the depth and width of the model, YOLOv5, which includes a backbone, neck, and head structure [66] similar to YOLOv4 [67], offers several model sizes. Therefore, it can be used on devices with a wide range of hardware specifications, including high and low computing resources. This adaptability guarantees that detection speed remains consistent, even on devices with restricted resources. Like monitoring tasks, the YOLOv5 model size variants can be tailored to match the individual data type, object class count, and hardware setup.

As a result, YOLOv5 enhances the monitoring system by serving as an object identifier. Contrary to YOLOv1 [61] and YOLOv2, which are referred to as YOLOv9000 [68], YOLOv3 [69] and YOLOv4 [67] provide two smaller variants called YOLOv3Tiny and YOLOv4Tiny. The YOLOv5 model comes in five different sizes: YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. Each size offers a distinct variation in performance and capabilities. The YOLOv5n model demonstrates higher speed but lower accuracy, whilst YOLOv5x boasts higher accuracy with little regard for speed. The approach proposed in this study is beneficial for computational processes that have restricted computing capabilities. Furthermore, it provides increased adaptability in choosing a model that is customized to the specific demands of the detection task, considering factors such as the amount and nature of the data.

D. 2D TO 3D PROJECTION

In the first phase of the 2D to 3D projection process, the pixel values that will be converted into a 3D LiDAR point cloud are analyzed. This study excludes intricate matters, as the pixel values in the 2D image are only obtained by calculating the 3D LiDAR point cloud coordinates. However, after precisely identifying the item, it is essential to do an object extraction procedure. This is due to the presence of unutilized point clouds that have not been included in the structure of the human object. To resolve this problem, clustering can be employed to guarantee that the final human object is completely devoid of contaminants. The initial stage in partitioning point cloud data into distinct clusters involves doing planar surface elimination on the unprocessed 3D LiDAR point cloud data. The flat plane is efficiently eliminated in both the vertical and horizontal directions through the use of histogram analysis along the normal vector. This is accomplished by utilizing the Singular Value Decomposition (SVD) technique on the Principal Component Analysis (PCA) plane created with kdTree-NN. Consequently, the point cloud exhibits a distinct clustering of things.

The method described in a previous study [70] can be employed to remove flat surfaces. This method builds

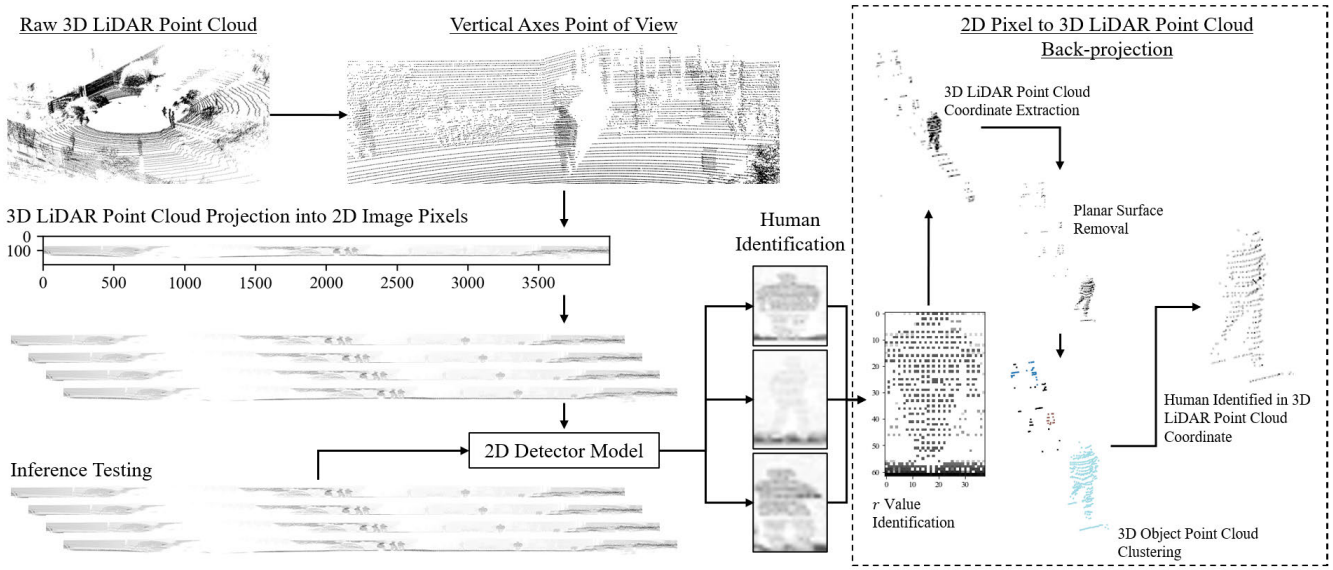

FIGURE 3. The process of applying a two-dimensional (2D) detector to identify humans represented in three-dimensional (3D) LiDAR point cloud data divided into several stages. The process begins with obtaining a Raw 3D LiDAR Point Cloud [24] and projecting a Vertical Axis Point of View onto a 2D plane, enabling the training and testing processes to be conducted in order to detect human objects, and finally performing the back-projection from identified human object in 2D image pixels onto 3D LiDAR point cloud coordinates.

upon the ground plane segmentation technique [49]. The investigation entails examining numerous sources on cluster formation, such as Random Sample Consensus (RANSAC) [71] and Density-based Spatial Clustering of Applications with Noise (DBSCAN) [72]. RANSAC is renowned for its robustness in dealing with noise and outliers in the dataset, which has solidified its reputation in the area. This approach has benefits in terms of computational efficiency and ease of implementation. Nevertheless, the RANSAC method is susceptible to parameter selection, which could restrict its efficacy when used on intricate datasets. DBSCAN, on the other hand, is a clustering method that effectively handles noise, accommodates clusters with various shapes, and does not require prior information on the expected number of clusters. However, it is susceptible to imprecise parameter values, which may lead to suboptimal outcomes when applied to datasets with unequal distribution.

III. PROPOSED METHODOLOGY

As found in several previous studies, 3D detection methods provide fidelity and accuracy in processing point clouds. However, the high computational and memory demands of managing large, complex point clouds present scalability challenges. On the other hand, a more computationally feasible approach is to project 3D data onto 2D planes, capitalizing on the effectiveness of pre-existing hardware and software that are specifically designed to process 2D data. Therefore, this study aims to investigate the utilization of single-stage 2D detector on 3D LiDAR point cloud data to decrease computational expenses. The focus is on identifying humans as the objects of interest. Due to the unsuitability of the BEV methodology for capturing human gestures, this study employed a 2D object detection method that utilized 3D LiDAR point cloud data projected onto a cylindrical plane. This process involved multiple stages which initially started with the generation of raw 3D point cloud data from the front perspective of the LiDAR. Subsequently, the 3D data which had cylindrical coordinates was transformed onto 2D image pixel coordinates by excluding the non-reflective part of the item.

After the 2D pixel data was gained, it was labelled by annotating the ground truth bounding B_{gt} with the YOLO label. This annotating stage was required to help the testing and evaluation of the proposed method using *mAP*. Lastly, the 2D image data was extracted and projected back to 3D LiDAR point cloud. This step is required to enable 3D human point cloud data analysis for further applied research such as Human Activity Monitoring. Figure 3 illustrates the overview of stages utilized in the proposed methodology.

A. 3D LIDAR POINT CLOUD DATA PREPARATION

During the stage of preparing 3D LiDAR point cloud data, the LiDAR system collected human gestures from the front perspective, resulting in the acquisition of 3D point cloud data. The resulting point cloud had different features, one of which was a hollow cylindrical shape. Therefore, the LiDAR laser emitted light in a way that specifically targeted surfaces capable of reflecting it, while ignoring the non-reflective surfaces of objects. The point cloud was obtained from the KITTI raw dataset. It consists of n points, represented as $Q = q_1, q_2, \ldots, q_n$, where each $q_i \in \mathbb{R}^3$. The point cloud can be represented as a matrix $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n]^T$, where each $\mathbf{q}_i$ is a vector with three components: q_{i_x}, q_{i_y}, and q_{i_z}. The p reflectance value was disregarded as the only criterion

was the point distance, which is determined by calculating the Euclidean distance between the x and y coordinates of the point and those of the LiDAR device, with $\mathbf{q}_l$ values $= [q_{l_x}, q_{l_y}, q_{l_z}]^T$. The point cloud data is denoted by the Eq. (1).

$$\mathbf{Q} = \begin{bmatrix} q_{i_{x_1}} & q_{i_{y_1}} & q_{i_{z_1}} \\ q_{i_{x_2}} & q_{i_{y_2}} & q_{i_{z_2}} \\ \vdots & \vdots & \vdots \\ q_{i_{x_n}} & q_{i_{y_n}} & q_{i_{z_n}} \end{bmatrix} \tag{1}$$

B. 3D CYLINDRICAL PLANE COORDINATE TO 2D PIXEL COORDINATE PROJECTION

The data, which had a cylindrical planar shape, was transformed into a 2D pixel space by excluding the non-reflective part of the item. Therefore, this approach required using the normalized z-axis $\in \mathbb{R}^3$ as a row $v \in \mathbb{R}^2$, which belongs to the set of real numbers $\mathbb{R}^3$, as a row $v \in \mathbb{R}^2$. In addition, by applying the angle θ and calculating $arctan\frac{q_{i_{y_i}}}{q_{i_{x_i}}}$, the x-axis and y-axis were used to generate a 2D column vector $u \in \mathbb{R}^2$. The r values were calculated using the formula $\sqrt{{q_{i_{x_i}}}^2 + {q_{i_{y_i}}}^2}$ and used in $\mathbf{Q}'$, which included u_i,v_i elements representing the spatial separation data between each point and the q_{l_x} and q_{l_y} coordinates of the LiDAR system. The results of 3D Cylindrical Plane Coordinate to 2D Pixel Coordinate Projection are expressed in Eq. (2).

$$\mathbf{Q}' = \begin{bmatrix} u_1 & v_1 & r_1 \\ u_2 & v_2 & r_2 \\ \vdots & \vdots & \vdots \\ u_n & v_n & r_n \end{bmatrix} \tag{2}$$

The 3D projection's coordinates were expressed in a 2D space using the equations $u = \theta$, $v = q_{i_{z_i}}$, where the values of u and v were determined by r. This study employed the normalized input value of z for row v [73]. The objective of this stage was to align the perspective of the 2D pixel space with the midpoint of the generated 2D pixel space in a perpendicular manner. Consequently, the v row was populated using the normalized z coordinates of each point in three-dimensional space. The projection procedure entailed mapping the complete 3D point cloud onto a 4000 × 200 2D pixel space, as depicted in Figure 4.

C. HUMAN DATA ANNOTATIONS FOR 2D DETECTOR MODEL TRAINING AND VALIDATION

Subsequently, the 2D image data was labeled by annotating the ground truth bounding box B_{gt} with the YOLO label in the format "*class*, u_{center}, v_{center}, *width*, and *height*". As a result, each image is given a particular item category and a standardized bounding box dimension that falls within the range of 0 to 1. Subsequently, the data was partitioned into three subsets, comprising 60% for training, 20% for validation, and 20% for testing. The YOLOv5 configuration was utilized as the detector in the current phase.

D. 2D DETECTOR TESTING AND EVALUATION

The study utilized the mean average precision (*mAP*) to evaluate the suggested detector since it is commonly used to assess the accuracy of object detection methods [74]. The *mAP* was deliberately chosen to support future investigations on multiclass detectors for human activity monitoring. However, the focus of object detection in the present study was limited to human objects.

The *mAP* was calculated by computing the average precision (*AP*), a widely used method in various object identification approaches showcased in the PASCAL VOC challenge [75], including R-CNN, Fast R-CNN, Faster R-CNN, YOLO, and SSD. *AP* values are calculated across a continuum of recall values ranging from 0 to 1. The *mAP* was utilized to calculate the average value of the data and make a larger number of forecasts. The *AP* utilized the confusion matrix, consisting of true positive (*TP*), true negative (*TN*), false positive (*FP*), and false negative (*FN*) values, to evaluate the accuracy of prediction models. To determine the values of *TP*, *TN*, *FP*, and *FN*, the Intersection over Union (*IoU*) measure was used as a benchmark. The *mAP* was calculated by evaluating the *IoU* between the predicted object's bounding box B_p and the ground truth bounding box B_{gt}, as shown in Eq. (3).

$$IoU = \frac{B_p \text{ and } B_{gt} \text{ overlapping area}}{B_p \text{ and } B_{gt} \text{ union area}} \tag{3}$$

Furthermore, the *IoU* outcomes were employed as benchmark values to obtain the precision (*P*) and recall (*R*), as illustrated in Eq. (4) and (5).

$$P = \frac{TP}{TP + FP} \tag{4}$$

$$R = \frac{TP}{No.\ of\ B_{gt}} \tag{5}$$

Subsequently, the *P* value for each *R* level was estimated by identifying the highest *P* value at which the *R* value surpassed the preceding *R*, as demonstrated in Eq. (6).

$$P_{interp}(R) = \max_{P:P \geq R} P(\tilde{R}) \tag{6}$$

Then, the *AP* value was ascertained by computing the mean precision value from the precision-recall (*PR*) curve, with recall values ranging from 0 to 1 in increments of 0.1. Eq. (7) demonstrates the computation of the *AP* value.

$$AP = \frac{1}{11} \sum_{R \in (0, 0.1, \ldots, 1)} P_{interp}(R) \tag{7}$$

After determining the *AP* for each class, the *mAP* value was then calculated. The *mAP* was computed using the formula $mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i$, where N represents the total number of classes. In relation with the model development, PASCAL VOC altered the regulations [76] regarding interpolated precision to include all recall values in a continuous manner, resulting in an unlimited range of recall values from 0 to 1. Meanwhile, the Microsoft Common Objects in Context

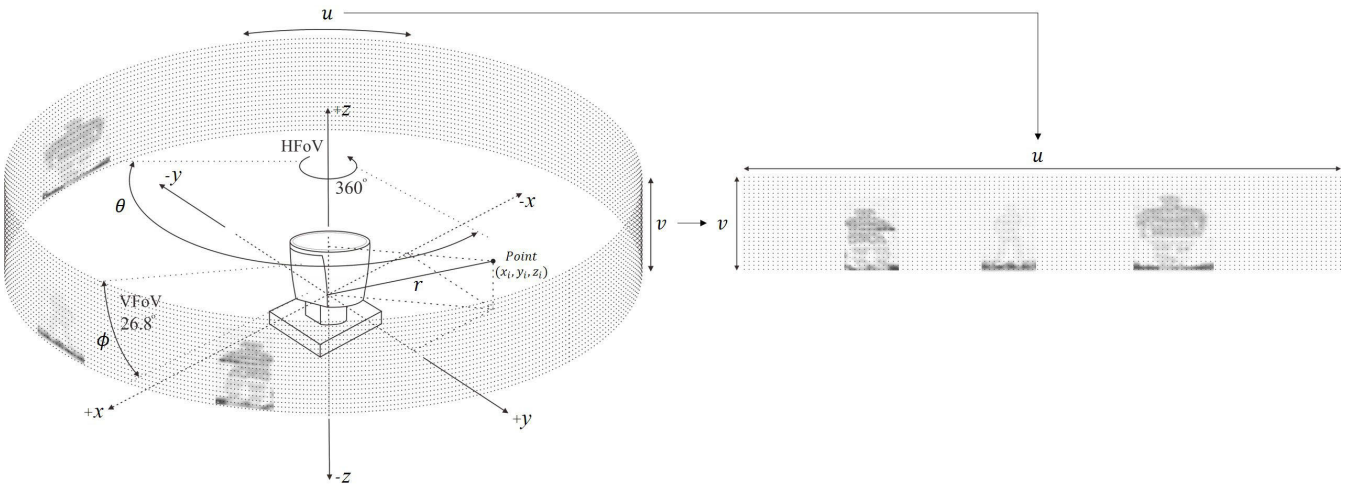

FIGURE 4. The process of projecting a 3D cylindrical plane onto a 2D image pixels that employed to extract human gesture features in the detection phase. This approach utilizes the cylindrical field of view (FoV) features formed by the range of the 360° 3D LiDAR beam to be projected onto a 2D pixel space.

(COCO) [77] utilized the 101-point recall rule to calculate interpolated precision.

The model evaluation approach utilized loss calculations based on the sum-squared error, in addition to the *mAP*. The reference values were obtained by comparing the ground truth data with the predictions. The YOLO loss calculations included the box loss, which represented the imprecision in localization, the object loss, which indicated the level of confidence, and the classification loss [61]. Eq. (8) denotes the YOLO loss function,

$$\begin{aligned} Loss = {} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} [(u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2] \\ & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} [(w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} (C_i - \hat{C}_i)^2 \\ & + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{noobj} (C_i - \hat{C}_i)^2 \\ & + \sum_{i=0}^{S^2} \mathbb{1}_{i}^{obj} \sum_{(c \in classes)} (P_i(c) - \hat{P}_i(c))^2 \end{aligned} \tag{8}$$

where S represents the cell grid used to predict the bounding box B, which is characterized by the coordinates u, v, w, and h. If the j^{th} bounding box in cell i detected the object, the $\mathbb{1}_i^{obj}$ value was 1; otherwise, it was 0. The λ_{coord} parameter was utilized to amplify the significance of the bounding box coordinate loss. Additionally, C_i indicated the confidence score of box j in cell i, whereas λ_{noobj} showed the weight of the loss when the background was identified. Furthermore, $P_i(c)$ denoted the probability of class c given the condition in cell i. The YOLOv5 loss function incorporates the previous YOLO loss function by utilizing the complete *IoU* (*CIoU*) [78] to maximize the regression bounding box loss. Moreover, the YOLOv5 loss algorithm incorporated both the object detection loss and classification loss.

E. 2D PIXEL TO 3D LIDAR POINT CLOUD OBJECT BACK-PROJECTION

The use of 2D to 3D back projection in this study was crucial in developing a computer vision system that employs a 2D detector implementation to detect objects, as it entailed a 3D data analysis. In order to initiate the implementation of this technique, the initial step included extracting a bounding box that precisely defined the location of elements in a two-dimensional image. This was achieved using the detection data produced by the YOLOv5 algorithm. Subsequently, the image's pixel data within the specified bounding box was examined to extract the r value of each pixel. The value was derived by computing the 3D LiDAR point cloud coordinates of the object and subsequently projecting them onto the image. During the third phase, sites with coordinates that matched the previously predicted r value were chosen. Subsequently, a ground plane elimination procedure was conducted to eliminate the flat surface in the image, yielding a more precise depiction of the object's relative position. The ground plane removal strategy utilized the outcomes of prior research by using the discoveries of normal vector analysis through the application of SVD on the PCA plane generated by kdTreeNN [49], [70].

The point cloud coordinates can be established by calculating the r value, which is obtained from the equation $\sqrt{{q_{i_{x_i}}}^2 + {q_{i_{y_i}}}^2}$. The value of $q_{i_{z_i}}$ could then be obtained by considering the corresponding $q_{i_{x_i}}$,$q_{i_{y_i}}$. As a result, a new

matrix $\mathbf{R}$ was generated, containing the x, y, and z coordinates of the 3D LiDAR human point cloud. This was accomplished by employing the correspondence data between the r value of the image pixels within the bounding box and the point cloud coordinates, as outlined in Eq. (9).

$$\mathbf{R} = \begin{bmatrix} r_1 & q_{i_{x_1}} & q_{i_{y_1}} & q_{i_{z_1}} \\ r_2 & q_{i_{x_2}} & q_{i_{y_2}} & q_{i_{z_2}} \\ \vdots & \vdots & \vdots & \vdots \\ r_n & q_{i_{x_n}} & q_{i_{y_n}} & q_{i_{z_n}} \end{bmatrix} \tag{9}$$

Furthermore, the planar surface removal operation is performed by leveraging the characteristics of the normal vector direction. To acquire a normal vector perpendicular to the PCA plane generated by kdTree-NN, the direction of the normal vector was initially identified using SVD. The PCA plane consisted of a group of nearby data points that encircled the $\mathbf{q}_i$ point, represented by the $\mathbf{R}$ elements and described by the equation $R' = r'_{i_1}, r'_{i_2}, \ldots, r'_{i_k}$. In this case, the q_i points were part of this collection and r'_{i_j} was distinct from q_i. The data structures $\mathbf{R'}_i$, consisting of $\left[\mathbf{r'}_{i_1}, \mathbf{r'}_{i_2}, \ldots, \mathbf{r'}_{i_k}\right]^T$, and $\mathbf{R'}_i^+$, consisting of $\left[\mathbf{q}_i, \mathbf{r'}_{i_1}, \ldots, \mathbf{r'}_{i_k}\right]^T$, were generated using the kdTree-NN approach. In this instance, the collection of adjacent points was denoted as $\mathbf{R'}_i$, while the collection that encompassed point $\mathbf{q}_i$ and its adjacent points $\mathbf{R'}_i$ was denoted as $\mathbf{R'}_i^+$. The mean vector of the coordinates $n_{i_x}, n_{i_y}, n_{i_z}$ was calculated with the SVD technique to the PCA $\mathbf{R'}_i^+$ plane. It involved subtracting the mean value from the PCA data matrix to reduce the level of variability. The revised data matrix was subsequently subjected to SVD, as indicated by Eq (10).

$$\min_{\mathbf{n}_i} \parallel \mathbf{A} \cdot \mathbf{n}_i \parallel_2 \tag{10}$$

$\mathbf{A}$ is the result of subtracting the vector $\mathbf{R'}_i^+$ from the mean vector $\bar{\mathbf{r}'}_i^+$ of $\mathbf{R'}_i^+$, which is obtained by averaging $\frac{1}{n}\sum_{i=1}^{n} \mathbf{R'}_i^+$. Given that SVD yields a distinct solution, it consequently generates the normal vector that is perpendicular to the plane formed by matrix $\mathbf{A}$. Therefore, it was justified to apply SVD to the modified matrix $\mathbf{A}$. After standardising the normal coordinates of the vector, each normal vector, $\mathbf{n}_i$, was divided by the magnitude of the vector, denoted as $|\mathbf{n}_i|$, which was calculated as $\sqrt{n_{i_x}^2 + n_{i_y}^2 + n_{i_z}^2}$. Therefore, the outcome was the unit vector, represented as $\hat{\mathbf{n}}_i$. A novel data format was devised to depict the coordinates of each point and its corresponding vector normal in the $\mathbf{S}$ matrix during the estimation phase of the surface normal vector, as produced in Eq. (11).

$$\mathbf{S} = \begin{bmatrix} q_{i_{x_1}} & q_{i_{y_1}} & q_{i_{z_1}} & \hat{n}_{i_{x_1}} & \hat{n}_{i_{y_1}} & \hat{n}_{i_{z_1}} \\ q_{i_{x_2}} & q_{i_{y_2}} & q_{i_{z_2}} & \hat{n}_{i_{x_2}} & \hat{n}_{i_{y_2}} & \hat{n}_{i_{z_2}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ q_{i_{x_n}} & q_{i_{y_n}} & q_{i_{z_n}} & \hat{n}_{i_{x_n}} & \hat{n}_{i_{y_n}} & \hat{n}_{i_{z_n}} \end{bmatrix} \tag{11}$$

Following that, the normal vector's direction was used to extract features. Various types of normal vectors could be utilized as reference features for eliminating locations in the planar plane due to their perpendicularity to the surface. A unit vector coordinate axis value close to 1 or 0 at point q_i represented a normal vector that was perpendicular to the y, z, x, z, or x, y planes among the several types of normal vectors. Initially, a check was performed to determine the direction of the normal vector. Subsequently, segmentation was carried out using a histogram with 20 divisions for each 0.05 increase within the range of 0 to 1 on the x, y, and z axes. The aim of this experiment was to establish that vector directions indicating flat surfaces were mostly seen to be either 0 or 1. This study proposes the use of a planar surface based on the presence of diverse coordinates. Consequently, elimination was performed in the histogram region that contained a substantial amount of data points.

After eliminating the planar planes, cluster formations were generated by categorising the clusters according to their density, which was determined by measuring the distance between each point. The DBSCAN method's implementation was driven by its simplicity and effectiveness at this stage [79]. Moreover, when dealing with extensive datasets, the DBSCAN algorithm is highly efficient in terms of computational performance. It is capable of processing such data in a relatively little timeframe [72]. The DBSCAN technique facilitated the grouping of three-dimensional LiDAR point cloud data. The clouds were initially partitioned using a planar surface. The outcomes of this phase are depicted in matrix $\mathbf{S}'$, as indicated in Eq. (11). The label l of the point cluster is denoted by the fourth column, while the point coordinates are denoted by the first three columns of the matrix.

$$\mathbf{S}' = \begin{bmatrix} q_{i_{x_1}} & q_{i_{y_1}} & q_{i_{z_1}} & l_1 \\ q_{i_{x_2}} & q_{i_{y_2}} & q_{i_{z_2}} & l_2 \\ \vdots & \vdots & \vdots & \vdots \\ q_{i_{x_n}} & q_{i_{y_n}} & q_{i_{z_n}} & l_n \end{bmatrix} \tag{12}$$

To generate the 3D LiDAR point cloud image of the identified object, the detected points were grouped together into clusters. The cluster with the greatest number of points was determined by analysing the labels. The techniques presented in this work can be utilised to detect and extract human 3D LiDAR point cloud data using a 2D detector. This data can then be returned to the 3D LiDAR point cloud, allowing for the acquisition of the coordinates of the human 3D LiDAR point cloud.

IV. EXPERIMENTAL RESULTS

The objective of this study is to streamline the computational process of object detection by utilizing 2D detectors on 3D LiDAR point cloud data. Since the proposed approach is specifically developed for future research on human activity analysis, we used humans as the objects of detection. This study employs the KITTI Raw Dataset 2011_09_28_drive_0208, which features a scenario with a human walking towards a LiDAR. There are 90000 3D

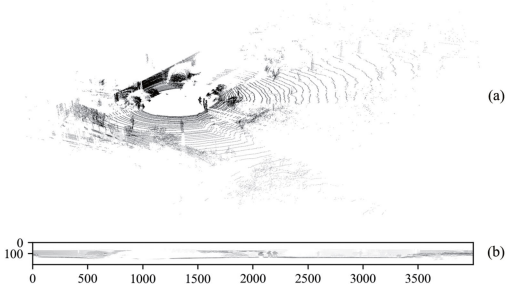

FIGURE 5. The projection of 3D cylinder plane coordinates to 2D image pixel coordinates has two components: (a) the generation of a 3D cylindrical plane from the Raw KITTI Dataset [24], and (b) the representation of this 3D space in a 2D image pixel.

point cloud data points transformed into a 2D image space with dimensions of 4000 × 200 pixels. The process consists of creating an empty matrix with dimensions of 4000 × 200 pixels, which was subsequently filled with a total of 90000 3D points. To map the x and y coordinates onto the u-axis which has a length of 4000 pixels, we employed variable θ. Similarly, the v-axis of 200 pixels was determined by normalizing the range of z values. Furthermore, the variable r was employed to populate a matrix consisting of 4000 columns and 200 rows with numerical values. The projection outcome is shown in Figure 5.

The next step was training the model using the default settings of the YOLOv5 training phase. The batch size was set to 16, the decay coefficient was 0.0005, the learning rate of the SGD optimizer was 0.001, and the training was performed for 400 iterations. The term ''batch size'' denotes the quantity of samples used for each training iteration. The decay parameter denotes the diminished significance at each iteration, which prevents overfitting. The learning rate governs how the training process progresses towards the optimal value. Furthermore, the overall number of iterations establishes the maximum threshold for the model's training frequency.

This study utilized the five variants of YOLOv5 detectors, namely YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, to identify the most effective model in terms of accuracy, speed, and model size. The *mAP* including *mAP*@0.5 and *mAP*@0.5:0.95, and the loss value including the box loss, object loss, and classification loss, were utilized for analysing the test results. Figure 6 illustrates the complete model's *mAP* outcomes.

The YOLOv5m, YOLOv5l, and YOLOv5x models attained exceptionally high *mAP*@0.5 scores, reaching a near-perfect value of 99.5%. However, YOLOv5s and YOLOv5n achieved lower *mAP* values of 99.21% and 98.23% correspondingly, at a threshold of 0.5. The data shows that the *mAP* at an *IoU* threshold of 0.5 for the five models has very little volatility and consistently remains high. In addition, YOLOv5x obtained the greatest *mAP* at the *IoU* threshold range of 0.5 to 0.95, with a score of 79.83%. YOLOv5l followed with a score of 76.88%, YOLOv5m

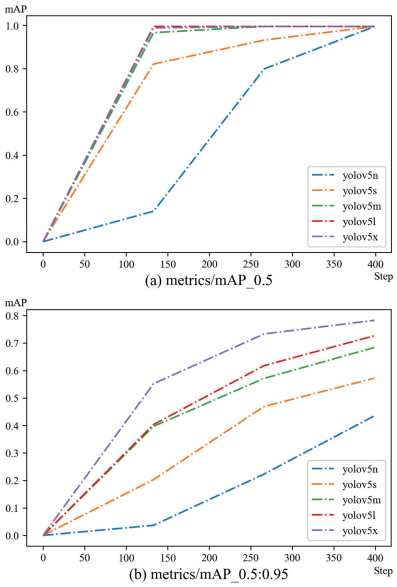

FIGURE 6. The comprehensive models' mean average precision (***mAP***) results provide promising indications of the effectiveness of the training outcomes. The evaluation metrics used in this study are: (a) mean Average Precision at an Intersection over Union (IoU) threshold of 0.5 (***mAP*@0.5**) and (b) mean Average Precision across IoU thresholds 0.5-0.95 (***mAP*@0.5:0.95**).

with 71.66%, YOLOv5s with 58.54%, and YOLOv5n with 44.35%. The diverse range of *mAP*@0.5:0.95 values indicates that YOLOv5x exhibits the highest level of accuracy. In addition, the YOLOv5l model achieved a *mAP*@0.5:0.95 that was only about 3 points lower than the YOLOv5x model. However, it is worth noting that the YOLOv5x model was 1.87 times larger than the YOLOv5l model.

Furthermore, the results were examined by considering the loss values, which encompass the box loss, object loss, and classification loss throughout both the model training and validation processes. The YOLOv5x model demonstrated a box loss of 0.027, which was reduced to 0.0092 throughout the validation process. The YOLOv5l, YOLOv5m, YOLOv5s, and YOLOv5n models displayed similar patterns, experiencing an average reduction of 51.43%. The current study also observes a significant decrease in box losses greater than 50% for YOLOv5x, YOLOv5l, and YOLOv5n. Nevertheless, the box loss results of all models are quite satisfactory, as the values are in close proximity to 0. Therefore, the model has the ability to precisely forecast the bounding box after undergoing the validation process. Furthermore, the YOLOv5x model demonstrated an object loss of 0.008 during the training phase, which then increased to 0.040 during the validation phase. The YOLOv5l, YOLOv5m, YOLOv5s, and YOLOv5n models displayed similar trends, with an average increase of 459.36%. YOLOv5n achieved the most object loss, measuring 0.012 during training and later increasing to 0.081 during validation. The classification loss continually remained at 0 as it only detected one class, particularly that

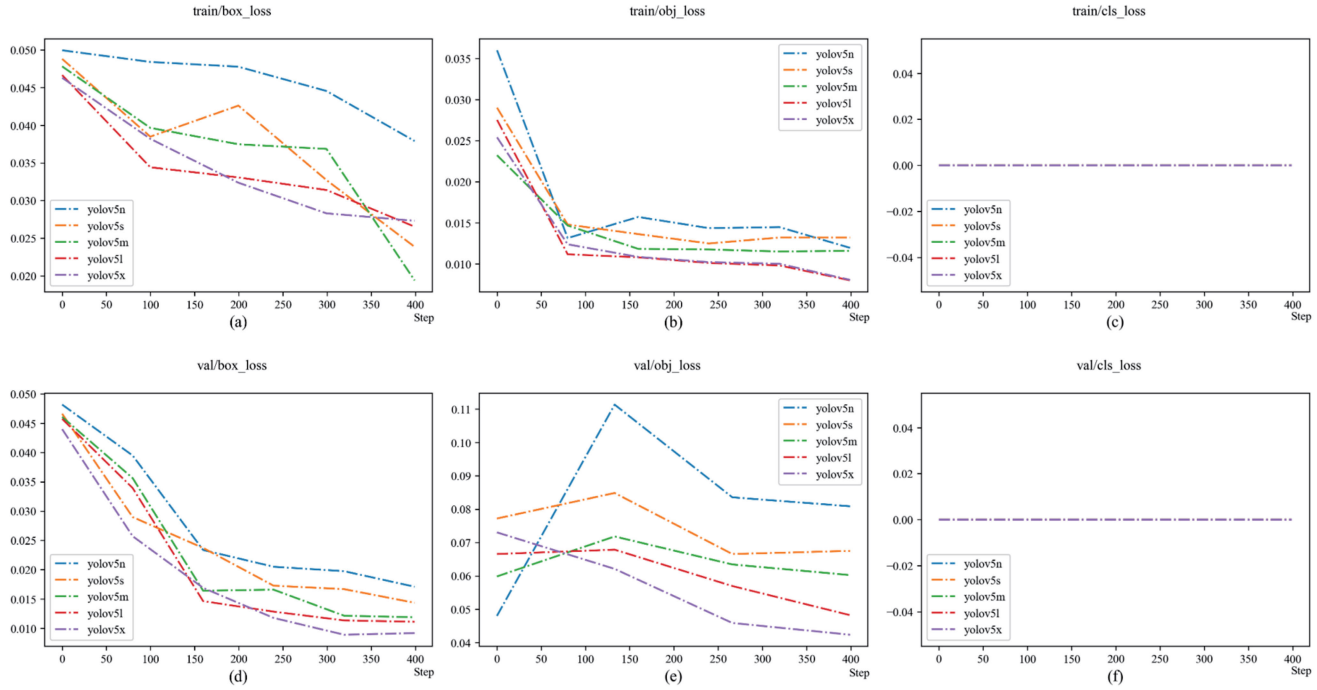

FIGURE 7. The outcomes of the model's loss evaluation encompass the following components: (a) the loss incurred by the bounding box during the training phase, (b) the loss associated with object detection during the training phase, (c) the loss incurred by classification during the training phase, (d) the loss incurred by the bounding box during the validation phase, (e) the loss associated with object detection during the validation phase, and (f) the loss incurred by classification during the validation phase.

of a human. Figure 7 presents the entire results of the loss analysis.

The models underwent later testing on an Nvidia Tesla T4 Graphics Processing Unit (GPU) and were produced via Google Colab. The test shows that the computing speed of the YOLOv5 model exhibits variability. The YOLOv5x model achieved an inference speed of 25 ms, the YOLOv5l model achieved an inference speed of 18 ms, the YOLOv5m model achieved an inference speed of 12.2 ms, the YOLOv5s model achieved an inference speed of 8.3 ms, and the YOLOv5n model achieved an inference speed of 7.6 ms. Figure 8 displays the results of the identification test. The data indicates that YOLOv5n significantly enhances the computing speed of the process for identifying human objects. Table 2 provides comprehensive information on the *mAP* test data, loss test data, and inference speed results obtained using the Nvidia Tesla T4.

After the completion of human object identification, the back-projection technique for the human object data is carried out during the inference phase. The back-projection process entails the mapping of the 2D pixel coordinates of the human object within the bounding box to the corresponding 3D LiDAR point cloud coordinates. Hence, the final result of the acquired person identification data is expressed as 3D LiDAR point cloud coordinates.

The back-projection approach in this study starts by analyzing the x and y coordinates used to get the r values that are then used to fill in the pixels in the 2D image. The correlation coefficients evaluated are specifically focused on the r values of pixels within the range of the bounding box. This guarantees that the acquired x and y values accurately correspond to items within the enclosing box. Moreover, upon obtaining the values of x and y, the z value may be derived as it immediately matches to the x and y values in the unprocessed 3D LiDAR point cloud data.

It is clear that a point cloud, which represents the background and flat plane features, is present and needs to be removed to obtain pure 3D LiDAR point cloud data. Therefore, the next step entails removing flat surfaces. The goal of planar surface removal is to eradicate 2D planes. The study implements planar surface removal by taking into account the normal direction of the vector that is orthogonal to the plane. This assumption is founded on the idea that points in a two-dimensional plane possess identical vector orientations.

After successfully eliminating flat surfaces, the next step involves clustering the items in the 3D LiDAR data point cloud based on their density. The current study used the DBSCAN technique to create clusters based on density. RANSAC, however, was not utilized due to its inadequate efficacy in processing 3D LiDAR point cloud data including intricate structures, such as overlapping or asymmetrical objects. The DBSCAN algorithm partitions data points into clusters based on their proximity, using a predefined distance threshold. Throughout the process of cluster development, many groupings of point clouds are generated, each sharing

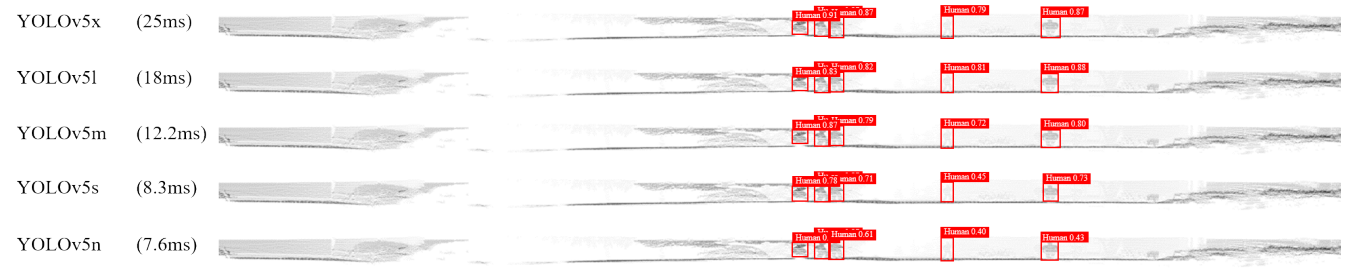

FIGURE 8. The results of human identification based on three-dimensional cylindrical plane coordinates that have been converted onto two-dimensional image pixels. The current approach makes use of a single-stage detector in conjunction with only 3D LiDAR point cloud data in order to effectively implement a vertical projection technique for the purpose of locating human things.

TABLE 2. YOLOv5 2D object detector model series applied on 3D LiDAR point cloud data test results.

YOLOv5 Model	Size (mb)	*mAP*@ 0.5(%)	*mAP*@0.5 :0.95(%)	Train (Box Loss)	Train (Obj. Loss)	Train (Cls. Loss)	Val. (Box Loss)	Val. (Obj. Loss)	Val. (Cls. Loss)	Inference Speed(ms)
YOLOv5x	172.9	99.50	79.83	0.027	0.008	0	0.0092	0.040	0	25
YOLOv5l	92.7	99.50	76.88	0.027	0.008	0	0.0100	0.046	0	18
YOLOv5m	42	99.50	71.66	0.019	0.012	0	0.0123	0.061	0	12.2
YOLOv5s	14.2	99.21	58.44	0.024	0.013	0	0.0152	0.070	0	8.3
YOLOv5n	3.7	98.23	44.35	0.038	0.012	0	0.0166	0.081	0	7.6

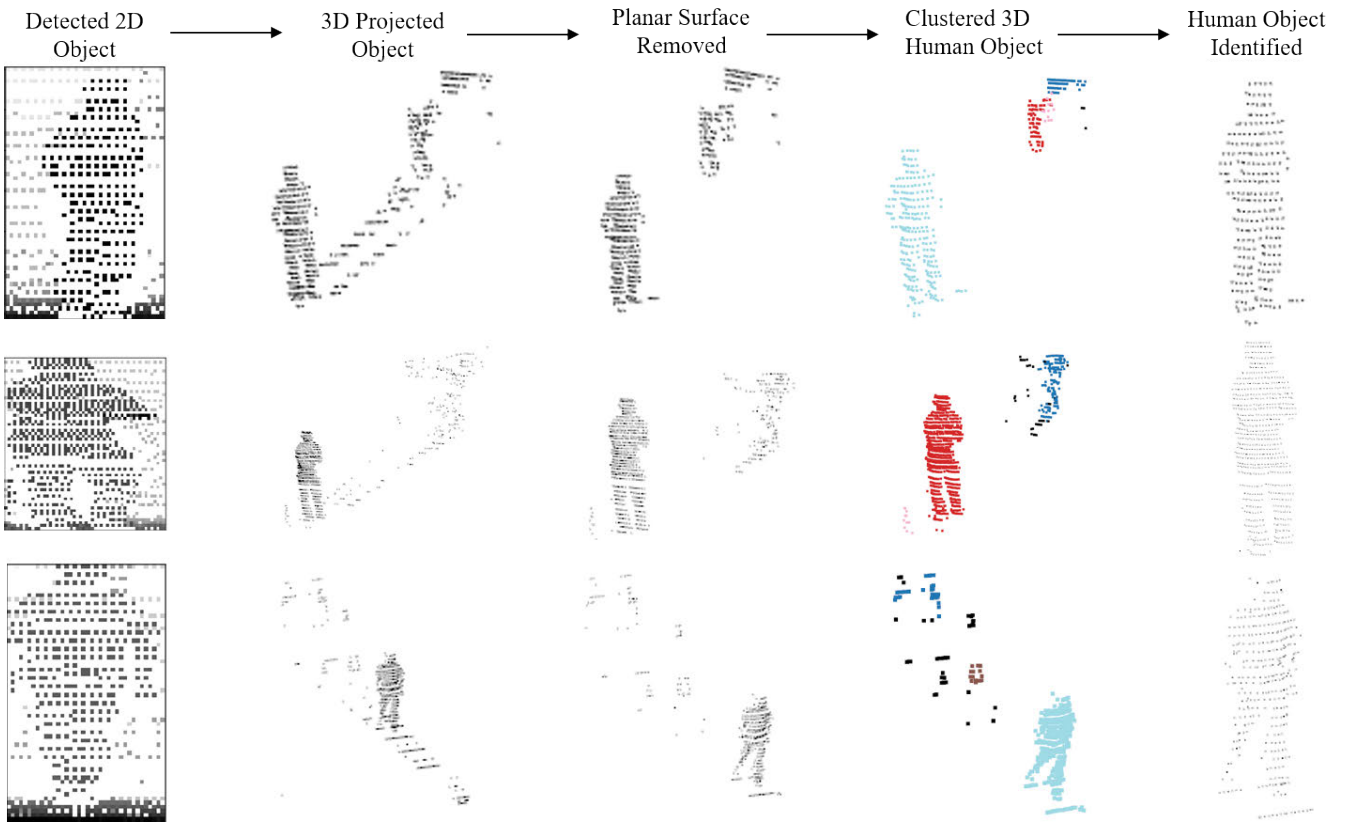

FIGURE 9. The outcome of back-projecting human objects, which have been recognized using bounding box 2D image pixels, onto 3D LiDAR point cloud coordinates. The methodology involves the identification of x, y, and z coordinates that correspond with the r value, subsequent elimination of planar surfaces, clustering of points using the DBSCAN algorithm, and finally sorting the clusters depending on their density.

the same cluster label. Consequently, points that have the same label are organized into a cluster that exhibits similarity.

The clustering procedure identifies the existence of clusters that have varying point densities. The investigation revealed that the identified human items exhibit a significant concentration and are situated at a substantial distance from the background, facilitating the creation of distinct clusters of objects. Hence, employing cluster selection based on point

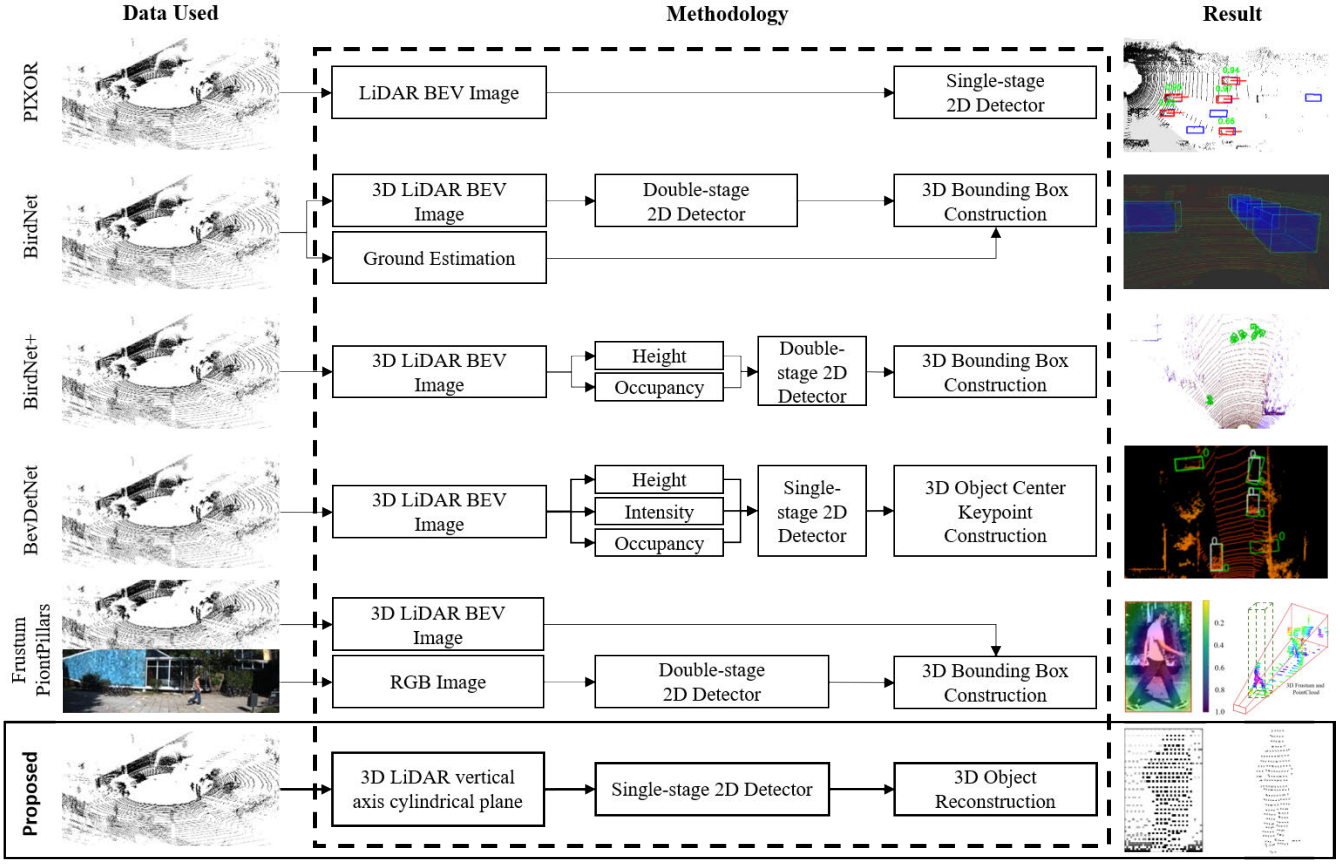

FIGURE 10. The proposed methodology contribution to the existing methodologies.

count can be employed to discover clusters that accurately depict human objects or entities. According to the data in this study, the method of identifying the cluster that best represents the item may be accomplished by arranging the number of points in each cluster in ascending order. The cluster with the highest number of points is then recognized as the representative cluster. Figure 9 illustrates the comprehensive outcomes of the 2D to 3D LiDAR back-projection for the identified human object. The study demonstrates that the utilization of a 2D projection-based method on 3D LiDAR point cloud data is an effective means of detecting and identifying human objects positioned in front of the sensor. Furthermore, the utilization of YOLOv5 can assist in choosing the most suitable model based on different computational capacities.

As mentioned, this method is proposed to enhance the existing techniques in processing and analyzing 3D LiDAR data. The result of this study shows that the proposed method is eligible to be an alternative in processing 3D LiDAR data through simplification and minimization. Other 3D LiDAR data processing methods can successfully identify human object. However, as the data is gained through BEV that lacks of vertical data and exposes unnecessary points, they may fail in presenting the object accurately. Meanwhile, the proposed method adopts Frustum-PointPillars that has been simplified by excluding RGB image. It also changes the BEV bounding box with clustering approach to gain object point cloud. Therefore, it can improve the accuracy and efficiency in the data process and analysis. Figure 10 further explains the contribution of the proposed method compared to the existing ones.

V. CONCLUSION

This work proposes an innovative approach to identify humans by utilizing unprocessed 3D LiDAR point cloud data, which is then analyzed using a single-stage 2D detector on projected vertical axis cylindrical coordinates. The projection technique relies on the characteristics of the 3D LiDAR point cloud, which has a cylindrical plane shape with the LiDAR device serving as the central axis. The current study utilized YOLOv5, a single-stage 2D object recognition framework, to choose the most optimal model based on factors such as model size, loss, accuracy, and inference speed. The YOLOv5 series has various YOLOv5x versions with varied model sizes, with YOLOv5n being the most compact type. The results of the model training and testing indicate that the overall model loss is suitable, with a value that is close to 0. The YOLOv5x model achieves the maximum accuracy, boasting an impressive *mAP* of 79.83% and an inference speed of 25 ms. In contrast, the YOLOv5n exhibits the least accuracy, achieving *mAP* of 44.35% and an inference speed of 7.6 ms. In addition, YOLOv5s surpasses YOLOv5n in

performance, achieving *mAP* of 58.54% and an inference speed of 8.3 ms. The YOLOv5l model exhibits *mAP* that is roughly 3 percentage points lower than that of YOLOv5x. Specifically, it achieves *mAP* value of 76.88% and an inference speed of 18 ms. In addition, the YOLOv5m model achieves *mAP* of 71.66% and has an inference speed of 12.2 ms.

All inference speeds were below 33.33 ms, indicating that objects can be detected before the LiDAR system begins its next azimuth rotation. The incorporation of a GPU into the monitoring process led to the acquisition of this result. YOLOv5x and YOLOv5l are likely to achieve positive results due to their high precision and low loss metrics, while also maintaining inference speeds of over 30 frames per second. Therefore, YOLOv5s and YOLOv5n can serve as substitutes when there are constraints on hardware resources, as these models have smaller computing footprints but can still achieve similar inference speeds. In addition, when considering intermediate computing specs, YOLOv5l and YOLOv5m could be a suitable option because to their similar accuracy to YOLOv5x and faster inference speeds compared to YOLOv5x.

Furthermore, during the inference phase, a successful inverse mapping of the identified object's 2D image pixels was performed to obtain the object's coordinates within the 3D LiDAR point cloud. Thus, this methodology has the capacity to function as a technique for identifying humans in monitoring procedures that utilize various computational equipment and make use of safe environmental data exploitation. To ensure privacy for individuals, it is possible to use just raw 3D LiDAR point cloud data. The raw 3D LiDAR point cloud data is processed through simplified Frustum-PointPillars with clustering approach. This can be employed to offer further assistance for subsequent research on surveillance systems that incorporate human monitoring, including inquiries on pedestrian behavior, walking rehabilitation, and other types of human activity monitoring.

ACKNOWLEDGMENT

The authors would like to thank the Indonesian Education Scholarship (BPI) Program of the Ministry of Education managed by Center for Higher Education (BPPT), in collaboration with the LPDP Scholarship Program of the Ministry of Finance, Indonesia; and the University Center of Excellence on Artificial Intelligence for Healthcare and Society (UCE AIHeS) Institut Teknologi Sepuluh Nopember, in collaboration with the Graduate School of Science and Technology, Kumamoto University, for fully supporting this work.

REFERENCES

[1] M. I. Perdana, W. Anggraeni, H. A. Sidharta, E. M. Yuniarno, and M. H. Purnomo, "Early warning pedestrian crossing intention from its head gesture using head pose estimation," in *Proc. Int. Seminar Intell. Technol. Appl.*, Jul. 2021, pp. 402–407, doi: 10.1109/ISITIA52817.2021.9502231.

[2] X. Chen, X. Luo, J. Weng, W. Luo, H. Li, and Q. Tian, "Multi-view gait image generation for cross-view gait recognition," *IEEE Trans. Image Process.*, vol. 30, pp. 3041–3055, 2021, doi: 10.1109/TIP.2021.3055936.

[3] I. Jarraya, F. B. Said, T. M. Hamdani, B. Neji, T. Beyrouthy, and A. M. Alimi, "Biometric-based security system for smart riding clubs," *IEEE Access*, vol. 10, pp. 132012–132030, 2022, doi: 10.1109/ACCESS.2022.3229260.

[4] M.-H. Nguyen, C.-C. Hsiao, W.-H. Cheng, and C.-C. Huang, "Practical 3D human skeleton tracking based on multi-view and multi-Kinect fusion," *Multimedia Syst.*, vol. 28, no. 2, pp. 529–552, Apr. 2022, doi: 10.1007/s00530-021-00846-x.

[5] S.-C. Hsu, Y.-W. Wang, and C.-L. Huang, "Human object identification for human–robot interaction by using fast R-CNN," in *Proc. 2nd IEEE Int. Conf. Robotic Comput. (IRC)*, Jan. 2018, pp. 201–204, doi: 10.1109/IRC.2018.00043.

[6] C. Vishnu, R. Datla, D. Roy, S. Babu, and C. K. Mohan, "Human fall detection in surveillance videos using fall motion vector modeling," *IEEE Sensors J.*, vol. 21, no. 15, pp. 17162–17170, Aug. 2021, doi: 10.1109/JSEN.2021.3082180.

[7] F. Lin, Z. Wang, H. Zhao, S. Qiu, X. Shi, L. Wu, R. Gravina, and G. Fortino, "Adaptive multi-modal fusion framework for activity monitoring of people with mobility disability," *IEEE J. Biomed. Health Informat.*, vol. 26, no. 8, pp. 4314–4324, Aug. 2022, doi: 10.1109/JBHI.2022.3168004.

[8] N. Hesse, S. Baumgartner, A. Gut, and H. J. A. van Hedel, "Concurrent validity of a custom method for markerless 3D full-body motion tracking of children and young adults based on a single RGB-D camera," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 31, pp. 1943–1951, 2023, doi: 10.1109/TNSRE.2023.3251440.

[9] F. Rosique, F. Losilla, and P. J. Navarro, "Using artificial vision for measuring the range of motion," *IEEE Latin Amer. Trans.*, vol. 19, no. 7, pp. 1129–1136, Jul. 2021, doi: 10.1109/TLA.2021.9461841.

[10] A. Naser, A. Lotfi, and J. Zhong, "Calibration of low-resolution thermal imaging for human monitoring applications," *IEEE Sensors Lett.*, vol. 6, no. 3, pp. 1–4, Mar. 2022, doi: 10.1109/LSENS.2022.3155936.

[11] M. Dumpis, D. Gedminas, and A. Serackis, "Inertial sensor based system for upper limb motion quantification," in *Proc. IEEE Open Conf. Electr., Electron. Inf. Sci.*, Apr. 2022, pp. 1–6.

[12] Y. Yu, S. Liu, Z. Zhang, Y. Liu, X. Liang, D. Li, L. Shuai, C. Wei, and L. Wei, "Far-field 3-D localization of radioactive hotspots via four-eyes stereo gamma camera," *IEEE Trans. Nucl. Sci.*, vol. 69, no. 8, pp. 1931–1938, Aug. 2022, doi: 10.1109/TNS.2022.3186433.

[13] H. Moradi, M. Karami, and S. Shamaghdari, "DeepSDP: A real-time deep stereo detection and positioning method for 3D object detection," in *Proc. 28th Iranian Conf. Electr. Eng. (ICEE)*. Iran: IEEE, Aug. 2020, pp. 1–5, doi: 10.1109/ICEE50131.2020.9260853.

[14] N. A. Ubina, S.-C. Cheng, C.-C. Chang, S.-Y. Cai, H.-Y. Lan, and H.-Y. Lu, "Intelligent underwater stereo camera design for fish metric estimation using reliable object matching," *IEEE Access*, vol. 10, pp. 74605–74619, 2022, doi: 10.1109/ACCESS.2022.3185753.

[15] A. Kuttner, M. Hauser, H. Zimmermann, and M. Hofbauer, "Highly sensitive indirect time-of-flight distance sensor with integrated single-photon avalanche diode in 0.35 μm CMOS," *IEEE Photon. J.*, vol. 14, no. 4, pp. 1–6, Aug. 2022, doi: 10.1109/JPHOT.2022.3182153.

[16] F. Piron, D. Morrison, M. R. Yuce, and J.-M. Redouté, "A review of single-photon avalanche diode time-of-flight imaging sensor arrays," *IEEE Sensors J.*, vol. 21, no. 11, pp. 12654–12666, Jun. 2021, doi: 10.1109/JSEN.2020.3039362.

[17] J. Mei and H. Zhao, "Incorporating human domain knowledge in 3-D LiDAR-based semantic segmentation," *IEEE Trans. Intell. Vehicles*, vol. 5, no. 2, pp. 178–187, Jun. 2020, doi: 10.1109/TIV.2019.2955851.

[18] B. Pal, S. Khaiyum, and Y. S. Kumaraswamy, "3D point cloud generation from 2D depth camera images using successive triangulation," in *Proc. Int. Conf. Innov. Mech. Ind. Appl. (ICIMIA)*, Feb. 2017, pp. 129–133, doi: 10.1109/ICIMIA.2017.7975586.

[19] S. Ko and S. Lee, "3D point cloud matching based on its 2D representation for visual odometry," in *Proc. IEEE Int. Conf. Image Process., Appl. Syst.*, Dec. 2018, pp. 216–219.

[20] H. Liu, K. Liao, C. Lin, Y. Zhao, and Y. Guo, "Pseudo-LiDAR point cloud interpolation based on 3D motion representation and spatial supervision," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6379–6389, Jul. 2022, doi: 10.1109/TITS.2021.3056048.

[21] J. P. Queralta, F. Yuhong, L. Salomaa, L. Qingqing, T. N. Gia, Z. Zou, H. Tenhunen, and T. Westerlund, "FPGA-based architecture for a low-cost 3D LiDAR design and implementation from multiple rotating 2D LiDARs with ROS," *IEEE SENSORS*, pp. 1–4, Oct. 2019.

[22] J. Roche, V. De-Silva, J. Hook, M. Moencks, and A. Kondoz, "A multimodal data processing system for LiDAR-based human activity recognition," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10027–10040, Oct. 2022, doi: 10.1109/TCYB.2021.3085489.

[23] Y. Li, L. Ma, Z. Zhong, F. Liu, M. A. Chapman, D. Cao, and J. Li, "Deep learning for LiDAR point clouds in autonomous driving: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3412–3432, Aug. 2021, doi: 10.1109/TNNLS.2020.3015992.

[24] 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, doi: 10.1177/0278364913491297.

[25] K. Zhang, S. Qiao, and K. Gao, "A new point cloud reconstruction algorithm based-on geometrical features," in *Proc. 7th Int. Conf. Model., Identificat. Control*, Dec. 2015, pp. 1–6, doi: 10.1109/ICMIC.2015.7409387.

[26] J. Tachella, Y. Altmann, N. Mellado, A. McCarthy, R. Tobin, G. S. Buller, J.-Y. Tourneret, and S. McLaughlin, "Real-time 3D reconstruction from single-photon LiDAR data using plug-and-play point cloud denoisers," *Nature Commun.*, vol. 10, no. 1, p. 4984, Nov. 2019, doi: 10.1038/s41467-019-12943-7.

[27] Y. Altmann, X. Ren, A. McCarthy, G. S. Buller, and S. McLaughlin, "LiDAR waveform-based analysis of depth images constructed using sparse single-photon data," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 1935–1946, May 2016, doi: 10.1109/TIP.2016.2526784.

[28] Q. Wang, Y. Tan, and Z. Mei, "Computational methods of acquisition and processing of 3D point cloud data for construction applications," *Arch. Comput. Methods Eng.*, vol. 27, no. 2, pp. 479–499, Apr. 2020, doi: 10.1007/s11831-019-09320-4.

[29] R. Volk, J. Stengel, and F. Schultmann, "Building information modeling (BIM) for existing buildings—Literature review and future needs," *Autom. Construct.*, vol. 38, pp. 109–127, Mar. 2014, doi: 10.1016/j.autcon.2013.10.023.

[30] Y. Wang, J. Cao, Y. Li, and C. Tu, "APM: Adaptive permutation module for point cloud classification," *Comput. Graph.*, vol. 97, pp. 217–224, Jun. 2021, doi: 10.1016/j.cag.2021.04.032.

[31] S. A. Bello, C. Wang, X. Sun, H. Deng, J. M. Adam, M. K. A. Bhatti, and N. M. Wambugu, "PDConv: Rigid transformation invariant convolution for 3D point clouds," *Expert Syst. Appl.*, vol. 210, Dec. 2022, Art. no. 118356, doi: 10.1016/j.eswa.2022.118356.

[32] Z. Zhang, J. Sun, Y. Dai, D. Zhou, X. Song, and M. He, "Self-supervised rigid transformation equivariance for accurate 3D point cloud registration," *Pattern Recognit.*, vol. 130, Oct. 2022, Art. no. 108784, doi: 10.1016/j.patcog.2022.108784.

[33] L. Bai, Y. Zhao, and X. Huang, "Enabling 3-D object detection with a low-resolution LiDAR," *IEEE Embedded Syst. Lett.*, vol. 14, no. 4, pp. 163–166, Dec. 2022, doi: 10.1109/LES.2022.3170298.

[34] A. Diab, R. Kashef, and A. Shaker, "Deep learning for LiDAR point cloud classification in remote sensing," *Sensors*, vol. 22, no. 20, p. 7868, Oct. 2022, doi: 10.3390/s22207868.

[35] B. Yang, W. Luo, and R. Urtasun, "PIXOR: Real-time 3D object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7652–7660, doi: 10.1109/CVPR.2018.00798.

[36] J. Beltrán, C. Guindel, F. M. Moreno, D. Cruzado, F. García, and A. De La Escalera, "BirdNet: A 3D object detection framework from LiDAR information," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3517–3523, doi: 10.1109/ITSC.2018.8569311.

[37] A. Barrera, C. Guindel, J. Beltrán, and F. García, "BirdNet+: End-to-end 3D object detection in LiDAR bird's eye view," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*. Greece: IEEE, Sep. 2020, pp. 1–6, doi: 10.1109/ITSC45102.2020.9294293.

[38] A. Barrera, J. Beltrán, C. Guindel, J. A. Iglesias, and F. García, "BirdNet+: Two-stage 3D object detection in LiDAR through a sparsity-invariant bird's eye view," *IEEE Access*, vol. 9, pp. 160299–160316, 2021, doi: 10.1109/ACCESS.2021.3131389.

[39] S. Mohapatra, S. Yogamani, H. Gotzig, S. Milz, and P. Mader, "BEVDet-Net: Bird's eye view LiDAR point cloud based real-time 3D object detection for autonomous driving," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 2809–2815.

[40] A. Paigwar, D. Sierra-Gonzalez, Ö. Erkent, and C. Laugier, "Frustum-pointpillars: A multi-stage approach for 3D object detection using RGB camera and LiDAR," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*. BC, Canada: IEEE, Oct. 2021, pp. 2926–2933, doi: 10.1109/ICCVW54120.2021.00327.

[41] Y. Zhang, Z. Xiang, C. Qiao, and S. Chen, "Accurate and real-time object detection based on bird's eye view on 3D point clouds," in *Proc. Int. Conf. 3D Vis. (3DV)*. QC, Canada: IEEE, Sep. 2019, pp. 214–221, doi: 10.1109/3DV.2019.00032.

[42] K. Zhao, L. Ma, Y. Meng, L. Liu, J. Wang, J. M. Junior, W. N. Gonçalves, and J. Li, "3D vehicle detection using multi-level fusion from point clouds and images," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15146–15154, Sep. 2022, doi: 10.1109/TITS.2021.3137392.

[43] S. Hou, Z. Wang, X. Li, and J. Song, "Three-dimensional multi-target tracking of point cloud from bird's-eye view," in *Proc. Int. Conf. Electron. Inf. Eng. Comput. Sci. (EIECS)*. China: IEEE, Sep. 2021, pp. 13–19, doi: 10.1109/EIECS53707.2021.9588028.

[44] S. N. Sridhara, E. Pavez, and A. Ortega, "Cylindrical coordinates for LiDAR point cloud compression," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*. AK, USA: IEEE, Sep. 2021, pp. 3083–3087, doi: 10.1109/ICIP42928.2021.9506448.

[45] J. Deng, W. Zhou, Y. Zhang, and H. Li, "From multi-view to hollow-3D: Hallucinated hollow-3D R-CNN for 3D object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 12, pp. 4722–4734, Dec. 2021, doi: 10.1109/TCSVT.2021.3100848.

[46] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4490–4499, doi: 10.1109/CVPR.2018.00472.

[47] Z. Yin, S. Pei, and Z. Yu, "End-to-end multi-view fusion for 3D object detection in LiDAR point clouds," in *Proc. Conf. Robot Learn.*, Osaka, Japan, 2019, pp. 923–932.

[48] M. Alsfasser, J. Siegemund, J. Kurian, and A. Kummert, "Exploiting polar grid structure and object shadows for fast object detection in point clouds," in *Proc. 12th Int. Conf. Mach. Vis.*, Jan. 2020, p. 27.

[49] N. Eka Budiyanta, E. Mulyanto Yuniarno, and M. Hery Purnomo, "Human point cloud data segmentation based on normal vector estimation using PCA-SVD approaches for elderly activity daily living detection," in *Proc. IEEE Region 10 Conf.*, Dec. 2021, pp. 632–636, doi: 10.1109/TENCON54134.2021.9707317.

[50] I. Alujaim, I. Park, and Y. Kim, "Human motion detection using planar array FMCW radar through 3D point clouds," in *Proc. 14th Eur. Conf. Antennas Propag. (EuCAP)*, Mar. 2020, pp. 1–3, doi: 10.23919/EuCAP48036.2020.9135381.

[51] S. Arshad, M. Shahzad, Q. Riaz, and M. M. Fraz, "DPRNet: Deep 3D point based residual network for semantic segmentation and classification of 3D point clouds," *IEEE Access*, vol. 7, pp. 68892–68904, 2019, doi: 10.1109/ACCESS.2019.2918862.

[52] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, Apr. 1980, doi: 10.1007/bf00344251.

[53] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese, "SEGCloud: Semantic segmentation of 3D point clouds," in *Proc. Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 537–547.

[54] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 922–928.

[55] W. Wang, R. Yu, Q. Huang, and U. Neumann, "SGPN: Similarity group proposal network for 3D point cloud instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2569–2578, doi: 10.1109/CVPR.2018.00272.

[56] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85, doi: 10.1109/CVPR.2017.16.

[57] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Sep. 2013, doi: 10.1007/s11263-013-0620-5.

[58] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[59] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*. Chile: IEEE, Dec. 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[60] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.
[61] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. NV, USA: IEEE, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.
[62] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Computer Vision*. Cham, Switzerland: Springer, 2016, pp. 21–37.
[63] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2874–2883.
[64] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2155–2162.
[65] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*. WA, USA: IEEE, Jun. 2020, pp. 10778–10787, doi: 10.1109/CVPR42600.2020.01079.
[66] G. Jocher et al., "Ultralytics/YOLOV5: V6.2–YOLOV5 classification models, apple M1, reproducibility, clearML and deci.AI integrations," *Zenodo*, Aug. 17, 2022.
[67] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOV4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
[68] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.
[69] J. Redmon and A. Farhadi, "YOLOV3: An incremental improvement," 2018, *arXiv:1804.02767*.
[70] N. E. Budiyanta, E. M. Yuniarno, T. Usagawa, and M. H. Purnomo, "Normal vector direction-based 3D LiDAR point cloud planar surface removal for object cluster minimization in human activity monitoring system," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf.*, May 2023, pp. 1–6, doi: 10.1109/I2MTC53148.2023.10175928.
[71] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981, doi: 10.1145/358669.358692.
[72] M. Ester, H.-P. Kriegel, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, Aug. 1996, pp. 226–231.
[73] Q. Chen, L. Sun, E. Cheung, and A. L. Yuille, "Every view counts: Cross-view consistency in 3D object detection with hybrid-cylindrical-spherical voxelization," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Dec. 2020, pp. 21224–21235.
[74] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.
[75] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010, doi: 10.1007/s11263-009-0275-4.
[76] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2015, doi: 10.1007/s11263-014-0733-5.
[77] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollar, "Microsoft COCO: Common objects in context," 2014, *arXiv:1405.0312*.
[78] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 7, pp. 12993–13000.
[79] D. Deng, "DBSCAN clustering algorithm based on density," in *Proc. 7th Int. Forum Electr. Eng. Autom. (IFEEA)*, Sep. 2020, pp. 949–953, doi: 10.1109/IFEEA51475.2020.00199.

NOVA EKA BUDIYANTA (Student Member, IEEE) received the bachelor's degree in mechatronics engineering education and the first master's degree in electrical engineering education from Universitas Negeri Yogyakarta, in 2013 and 2017, respectively, and the second master's degree from the Electrical Engineering Master Program, Universitas Katolik Indonesia Atma Jaya, in 2020. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember. He has experience in hardware–software programming. He is also a Lecturer with the Department of Electrical Engineering, Universitas Katolik Indonesia Atma Jaya, concerned with image processing, robotics, and machine learning research field.

EKO MULYANTO YUNIARNO (Member, IEEE) received the bachelor's, master's, and Ph.D. degrees in electrical engineering from Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia, in 1995, 2005, and 2013, respectively. Currently, he is a Senior Lecturer in computer engineering with the Department of Computer Engineering, ITS. His research interests include computer vision, image processing, machine learning, and deep learning.

TSUYOSHI USAGAWA (Member, IEEE) received the B.E. degree from Kyushu Institute of Technology, in 1981, and the M.E. degree from Tohoku University, in 1983. Since 1983, he has been with Kumamoto University, Japan, where he has been a Professor with the Graduate School of Science and Technology, since 2004. From 2014 to 2020, he was the Vice Dean of the Graduate School of Science and Technology, Kumamoto University. He is currently the Trustee Vice President of Kumamoto University. His research interests include acoustic signal processing, perceptual information processing, educational data mining, and e-learning. He is a member of ASA, ASJ, INCE/J, IEICE, JSET, JAIS, and ACM. From 2005 to 2007, he was the Vice President of ASJ.

MAURIDHI HERY PURNOMO (Senior Member, IEEE) received the B.S. degree from the Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia, in 1984, and the M.S. and Ph.D. degrees from the Department of Electrical Engineering, Osaka City University, Osaka, Japan, in 1995 and 1998, respectively. He is currently a Professor with ITS, and has involved in teaching research philosophy, artificial intelligence, neural networks, and image processing. His research interests include smart grids, renewable energy, and artificial intelligent application in healthcare and power systems. He is the Chair of the IEEE Industrial Electronics Society Indonesia Section.