Detection and Tracking in Human Monitoring Framework Using Modified Direct 3D LiDAR Point Cloud Classifier Based on Region Cluster Proposal

Nova Eka Budiyanta
Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember

Eko Mulyanto Yuniarno
Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember

Usagawa, Tsuyoshi
Graduate School of Science and Technology, Kumamoto University

Mauridhi Hery Purnomo
Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember

https://doi.org/10.5109/7236849

出版情報：Evergreen. 11 (3), pp.2022-2034, 2024-09. Interdisciplinary Graduate School of Engineering Sciences, Kyushu University, Japan
バージョン：

Detection and Tracking in Human Monitoring Framework Using Modified Direct 3D LiDAR Point Cloud Classifier Based on Region Cluster Proposal

Nova Eka Budiyanta[1,4], Eko Mulyanto Yuniarno[1,2], Tsuyoshi Usagawa[3], Mauridhi Hery Purnomo[1,2,5,*]

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

*Author to whom correspondence should be addressed:
E-mail: hery@ee.its.ac.id

(Received December 13, 2023: Revised May 7, 2024: Accepted July 21, 2024).

Abstract: Using camera pixel data in visual data monitoring raises privacy issues as it captures the entire environment and sensitive information. Hence, numerous studies have investigated human monitoring procedures using Three-dimensional Light Detection and Ranging (3D LiDAR), specifically focusing on detection and tracking tasks to mitigate potential health risks from positional patterns. Unfortunately, unprocessed 3D LiDAR point clouds are challenging to detect and track due to their dispersed nature. As suggested in many studies, restructuring strategies effectively decrease information loss but require higher computational costs. This study proposes the utilization of a direct point processing approach based on a region cluster proposal on a modified PointNet classifier as human detection and tracking framework. The modified PointNet classifier has demonstrated an improved accuracy of 98.79% in the classification process for supporting human object detection, higher than the default architecture which yields an accuracy of 94.98%. Furthermore, this study also develops a distance estimation technique to enhance the tracking process. In general, the human detection and tracking procedure demonstrates satisfactory performance through the utilization of a solitary data type, specifically the unprocessed 3D LiDAR point cloud, which is processed directly using a modified PointNet classifier.

Keywords: 3D LiDAR Point Cloud; Human Detection and Tracking Framework; Modified PointNet Classifier; Monitoring System; Region Cluster Proposal

1. Introduction

The monitoring system is a method employed to observe and track alterations in a given system. This process involves framework for collecting new data generated by the object under observation within a specific timeframe. Numerous studies have been undertaken pertaining to the advancement of a monitoring system. One area of study that garners significant attention and intrigue while also necessitating further investigation is the monitoring system for human objects. Comprehensive investigations of the system to facilitate monitoring in the domains of health[1,2], behavior[3,4], and activities[5,6] remain imperative. These studies encompass a diverse array of factors that impact monitoring outcomes.

Implementing human monitoring can facilitate the effective examination of changes in an individual's data. In the healthcare system, the monitoring process is important in determining the outcomes of diagnoses and drawing conclusions regarding the necessary actions to uphold the patients' wellbeing[7–9]. Moreover, the utilization of behavior monitoring enables to forecast the patients' inclination to engage in a specific behavior, as well as their predisposition toward particular circumstances[10]. Monitoring the human activities can also serve as a means of identifying anomalies[11], thereby mitigating their potential risks[12], that can be done by implementing detection and tracking process[13]. Additionally, it can also be employed to identify variations in gait during the patients' walking activity, potentially indicating a decline in their overall health.

Considering these rationales, a multitude of investigations have been conducted with the aim of developing frameworks for human monitoring. The human monitoring process in several frameworks can be conducted using wearable and non-wearable devices, as indicated by previous studies[14–17]. Wearable devices typically prioritize the monitoring of individuals' vital functions, including heart rate, oxygen levels, and body temperature. Additionally, many of these devices are equipped with Global Positioning System (GPS) technology to facilitate location tracking for monitoring purposes. Nevertheless, the utilization of wearable devices is highly susceptible to individual negligence and is constrained by its personal scope. If the individual being monitored fails to wear the wearable device, the monitoring process can impossibly be conducted. Conversely, non-wearable devices, predominantly employing visual concepts, may offer a potential resolution to this issue. The utilization of this visual-based monitoring, especially in detection and tracking tasks, RGB camera commonly employed to capture the bodily gestures of the subject under surveillance. As a result, this presents a significant concern regarding privacy infringement as it enables the comprehensive acquisition of data from the surrounding environment. Indeed, when constructing a surveillance system for human entities, it is essential to recognize the significance of privacy safety as a critical determinant.

Such potential violation of individuals' privacy urges further development of human detection and tracking framework to reduce the issues[18]. In this study, a non-wearable device with 3D LiDAR technology is employed as an alternative. The utilization of a 3D LiDAR point cloud offers greater advantages than a camera due to its ability to rapidly and accurately determine the three-dimensional coordinates of objects. This eliminates the need for employing photogrammetry techniques commonly used with cameras, which often yield suboptimal levels of accuracy[19]. Fig. 1[20] presents a comparative analysis of pixel data obtained from camera usage and alternative data provided by 3D LiDAR as a potential solution.

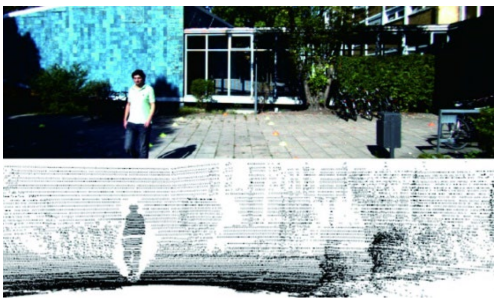
(a)
(b)


Fig. 1: Comparative reference data[20]:
(a) Environment captured in camera pixels,
(b) Environment captured in 3D LiDAR point cloud

The 3D LiDAR point cloud is generated by reflecting the LiDAR beam off the surface of the designated object, resulting in a representation of coordinates that can be visualized as human points. Several types of point coordinates are commonly generated, including cartesian coordinates[21,22] cylindrical coordinates[23], and spherical coordinates[22]. Hence, the spatial coordinates of the object are distinctly depicted in a 3D framework. While point clouds generated from 3D LiDAR technology have the potential to serve as an alternative means of reducing privacy concerns compared to camera pixels, their processing presents its own set of challenges. The reason for this discrepancy can be attributed to the unstructured nature of the point cloud data structure, as opposed to the organized framework of the camera pixel system. This phenomenon arises due to the inconsistency of the reflection plane for the laser beam in each cycle. Consequently, the coordinate points constituting the object exhibit variations within a single cycle of the azimuth rotation in 3D LiDAR[24]. Therefore, it is imperative to employ multiple methodologies to ensure that the utilization of 3D LiDAR point clouds yields significant insights, particularly within the context of the human detection and tracking framework examined in this study.

During the monitoring process, 3D LiDAR point clouds exhibit a relatively low density of points compared to other applications where a complete reflection of the object's surface is captured from all angles[25]. The utilization of a 3D LiDAR point cloud, which serves as a visual representation of an object's surface, leads to the inclusion of blank areas in the detection and tracking process. These blank areas are incorporated to ensure that objects lacking the ability to reflect the LiDAR beam are not depicted through coordinate points[26]. Hence, it is frequently observed that the resultant form of the object does not accurately depict its entirety. As a result, 3D LiDAR point cloud processing has emerged as a focal point in numerous academic investigations. The complexity of utilizing the point cloud data structure generated by 3D LiDAR in monitoring framework is a contributing factor to this phenomenon.

1.1. Motivation

Several studies have made various attempts to incorporate point clouds into monitoring frameworks[27,28]. Given the necessity for monitoring frameworks in detecting and tracking objects periodically to acquire optimal information, deep learning is a common method to process 3D LiDAR point cloud data[29] . However, most of the deep learning methods require structured data[30]. For instance, the multi-view approach incorporates several features such as front-view, bird-eye view (BEV), and image to obtain 3D rotates box. Hence, this approach requires relatively high computational cost.

One of the other common approaches to process 3D LiDAR with deep learning method is segmentation-based approach. This method uses semantic segmentation to remove the background points of the object which results on some foreground point proposals. Besides segmentation, frustum-based method is also common in 3D LiDAR processing. This method uses 2D detector to obtain 2D object candidate areas and extracts 3D frustum proposal to each of the 2D candidate areas. This method is efficient to propose possible 3D object locations but the step-by-step pipelines limits the process with 2D detector. All the methods mentioned above employ 3D LiDAR point cloud and image as the modality. Consequently, the computational cost is relatively high and it potentially decreases the processing speed.

In order to cut the cost and improve the speed, some researchers reduce the modality by using 3D LiDAR point cloud only. This strategy is commonly used in the projection-based, discretization-based, and direct point-based approach. The projection-based approach uses bird-eye-view (BEV) which projects the top-view point cloud data to discretize point cloud of grid 2D in the object detection. Meanwhile, discretization-based method converts point clouds into tensors and feeds them into a backbone network to extract multilevel features. This method requires some additional networks with point-level monitoring to study the point cloud structure. This step is also well-known as voxelization[31,32]. Therefore, in both projection and discretization methods, additional step such as convolution at 2D projection plane or voxelization on 3D plane is needed to detect the object and its location.

Meanwhile, some other studies employ direct processing to improve the efficiency. This method proceeds the point clouds without conversing the voxel or utilizing any data types other than point clouds. As a result, this method is highly effective in minimizing the processing time and decreasing computational expenses. Commonly, this method is based on influential direct point processing technique known as PointNet[33] which enables the utilization of 3D Point Cloud data as direct input for training Deep Learning models. It has been proven to be an effective and viable approach for 3D object classification in diverse Point Cloud data processing applications[34,35]. Figure 2 presents the comparison of various 3D LiDAR point cloud processing approaches.

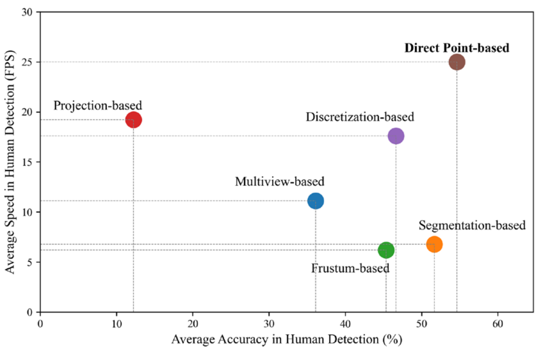


Fig. 2: Human detection approach results on the KITTI Dataset comparison[29].

1.2. Main Contribution

This study implements a novel human monitoring framework by incorporating PointNet, a deep learning model, to directly cluster regions of interest on 3D LiDAR point cloud-based proposals. This approach avoids voxelization and inclusion of other data types typically employed in similar studies which underwent preprocessing through the application of a planar surface removal approach. Subsequently, constructed clusters were formed using our preprocessing technique in former study[36]. The elimination of the flat plane results in the disconnection between objects, thereby enabling the construction of object clusters that can serve as input classifiers. Therefore, the present study utilizes a modified PointNet-based classifier to discern the distinction between human and non-human objects within a given cluster.

Broadly speaking, contributions are added to the development of the human monitoring framework by integrating detection and tracking work through this study in the following aspects:

- Region cluster proposal supporting procedures for implementing object detection in 3D LiDAR point cloud human detection and tracking framework.
- The application of a single 3D LiDAR point cloud data type in capturing human representation data as privacy preservation.
- Improved accuracy and loss values optimization in the modified PointNet-based classifier to support the detection process.
- Distance estimation procedure based on a bounding box to support object tracking in human monitoring.

The paper continues as follows: Section 2 discusses related works that support our contribution; Section 3 describes our proposed methods for object detection and tracking framework development as our contribution; Section 4 presents the results of the experimental phase; and Section 5 concludes with a summary and application considerations.

2. Related Work

2.1. 3D LiDAR Point Cloud as Reference Data

The projection approach, voxel generation, and raw point cloud data processing are a few of the several 3D LiDAR point cloud-based object detection methods. As in PIXOR[37], BirdNet[38], BirdNet+[39,40], and BevDetNet[41], adopting a bird-eyeview (BEV) level to obtain a top-down view of the 3D LiDAR point cloud data is a typical projection technique. By using data conversion from the BEV level into 2D picture space, the detector can handle point cloud data more quickly. However, when representing information from 3D space in 2D picture space, the projection approach frequently causes information from 3D space to be lost. Comparing the volumetric method with voxelization to the projection method in 2D space, on the other hand, offers a solution by minimizing the loss of information from 3D space. However, voxelization methods like VoxelNet[21], Frustum PointNet[42], and Multi-View Frustum Pointnet (MVFP)[43], employ generate sparse point clouds for sampling into voxels which can potentially remove information from the point cloud itself instead of minimizing the loss of information compared to projection approaches. Compared to the projection method into 2D space, the voxelization procedure also has substantial computational overhead, which may cause a relatively longer processing time. Approaches based on unprocessed 3D point clouds, including PointNet and its variations, evolved in response to this issue. Compared to the other two ways, the raw 3D LiDAR point cloud-based approach is thought to be better at keeping the point cloud's current information.

2.2. Cluster Reconstruction

In order to effectively utilize raw 3D LiDAR point cloud, the PointNet network requires input in the form of clustered points, which are representative of the object to be recognized. Hence, it is imperative to partition the dataset into distinct clusters of objects, which will subsequently serve as input data for PointNet. The initial step in partitioning point cloud data into distinct clusters involves implementing planar surface removal on the unprocessed 3D LiDAR point cloud data. The successful removal of the flat plane in both vertical and horizontal directions is achieved by employing histogram analysis along the normal vector, utilizing the Singular Value Decomposition (SVD)[44] approach to the Principal Component Analysis (PCA)[45,46] plane generated through kdTree-NN[47] so the point cloud exhibits a grouping of entities. The removal of planar surfaces is conducted using the methodology outlined in a prior study[36], which builds upon the ground plane segmentation technique described in an earlier study[26]. The investigation encompasses an examination of various references on cluster construction, such as Random Sample Consensus (RANSAC)[48] and Density-based Spatial Clustering of Applications with Noise (DBSCAN)[49]. RANSAC is renowned for its robustness in dealing with noise and outliers within the dataset, hence earning its reputation in the field. This solution provides benefits in terms of computational efficiency and ease of implementation. Nevertheless, the RANSAC algorithm demonstrates a certain degree of sensitivity to selecting its parameters, potentially constraining its effectiveness when applied to intricate datasets. In contrast, DBSCAN is a density-based clustering technique that effectively handles noise, accommodates clusters with varying shapes, and does not rely on prior knowledge of the expected number of clusters. However, it exhibits a high degree of sensitivity towards imprecise parameter values, which may lead to suboptimal outcomes when applied to datasets characterized by uneven density.

2.3. Direct Point Processing

PointNet is a pioneering neural network architecture in the field of point cloud data processing. It is specifically designed to process point data without requiring manual conversion or feature extraction by the users[50]. The PointNet architecture comprises multiple layers that collaboratively extract significant information from individual points constituting the input. The input transformation layer is employed to standardize and convert each endpoint to be insensitive to rotation and translation[51], utilizing T-Net[52]. As a result, the network has the capability to extract information that is independent of the absolute position or orientation of the constituent points comprising the network. Subsequently, a point transformation layer is employed to derive the localized attributes of individual points by taking into account the contextual information provided by the neighboring points. This layer employs learning functions to construct a distinct transformation matrix for each endpoint. Hence, the attributes of the region can be modified and adjusted to align with the prevailing conditions of the surrounding vicinity.

The max-pooling layer is essential in the PointNet architecture as it facilitates the aggregation of local point features, enabling the creation of a comprehensive global representation for the entire point cloud. Effectively addressing this concatenation requires extracting the maximum value from each feature dimension[53]. The provided global representation effectively depicts significant characteristics regarding the composition and properties of the complete point cloud. The ultimate phase entails transmitting this comprehensive representation across multiple interconnected layers to execute the classification or segmentation task. The distinct competitive advantage of PointNet lies in its ability to directly process point data and its flexibility in accommodating variations in point order and number[54]. Consequently, the collected point data enables the derivation of both global and local information.

3. Proposed Methodology

This study proposed a framework that was conducted in three distinct phases, each with a series of sequential procedures. The initial phase of this investigation was the process of cluster construction. During the initial phase, the process involved the execution of Raw 3D LiDAR Point Cloud Data Preparation, Planar Surface Removal, and Density-based Cluster Construction. This study's subsequent phase suggested clusters to be subjected to experimentation using the modified PointNet algorithm. During this phase, the execution of the procedure involved the utilization of the Region Point Cluster Proposal and Direct Point Cluster Classification processes. Additionally, the third phase of this study aimed to estimate bounding box-based distances for the segmented human object outcomes. The findings of this study pertained to the

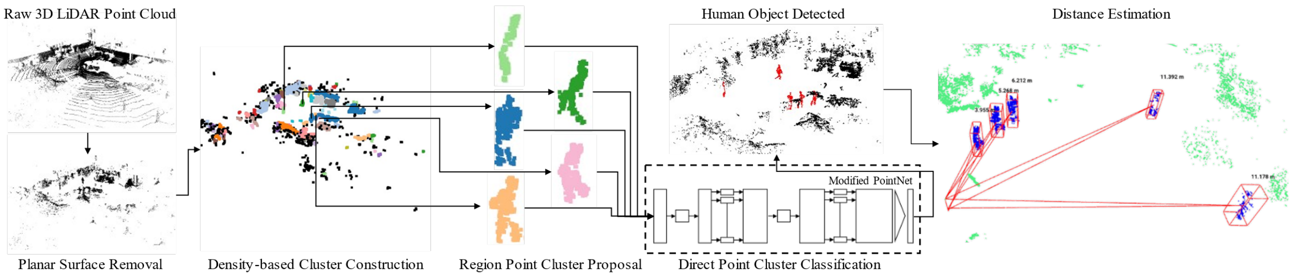


Fig. 3: Framework proposed in applying direct point processing for detection and tracking in human monitoring exploiting raw 3D LiDAR point cloud.

detection and tracking of humans in the point cloud representation acquired in conjunction with the estimated distance to the LiDAR position. Therefore, they have potential implications for the development and implementation of monitoring systems in human contexts. The framework proposed in this study can be seen in Fig. 3.

3.1. Raw 3D LiDAR Point Cloud Data Preparation

The initial stage in the first phase of this study began with raw 3D LiDAR point cloud preparation. As in previous studies, the raw 3D LiDAR point cloud used in this study was taken from the KITTI raw dataset, which was structured in cartesian coordinates. Represented in the $\mathbf{P}$ matrix, the dataset consisted of a number of n points with notation $\{p_1, p_2, \dots, p_n\}, p_i \in \mathbb{R}^3$, with $p_i = \left[p_{i_x}, p_{i_y}, p_{i_z}\right]$. Thus, the raw 3D LiDAR point cloud data structure used in this study is represented in Eq. (1).

$$\mathbf{P} = \begin{bmatrix} p_{i_{x_1}} & p_{i_{y_1}} & p_{i_{z_1}} \\ p_{i_{x_2}} & p_{i_{y_2}} & p_{i_{z_2}} \\ \vdots & \vdots & \vdots \\ p_{i_{x_n}} & p_{i_{y_n}} & p_{i_{z_n}} \end{bmatrix} \quad (1)$$

3.2. Planar Surface Removal

This study implemented a previously validated methodology to substantiate the planar surface removal procedure that leverages the characteristics of the normal vector direction. The determination of the normal vector direction was initially achieved through SVD to derive a normal vector that was orthogonal to the PCA plane generated using kdTree-NN. At this juncture, the PCA plane encompassed a cluster of adjacent data points surrounding the $\mathbf{p}_i$ point, denoted as $Q = q_{i_1}, q_{i_2}, \dots, q_{i_k}$, which represented the P elements. In this case, q_{i_j} was distinct from p_i and the p_i points were included in this set. The data structures $\mathbf{Q}_i = \left[\mathbf{q}_{i_1}, \mathbf{q}_{i_1}, \dots, \mathbf{q}_{i_k}\right]^T$ and $\mathbf{Q}_i^+ = \left[\mathbf{p}_i, \mathbf{q}_{i_1}, \dots, \mathbf{q}_{i_k}\right]^T$ were generated through the utilisation of kdTree-NN algorithm. In this particular scenario, $\mathbf{Q}_i$ denoted the set of adjacent points, whereas $\mathbf{Q}_i^+$ denoted the set consisting of point $\mathbf{p}_i$ and its adjacent points $\mathbf{Q}_i$. Subsequently, the SVD technique was employed on the PCA $\mathbf{Q}_i^+$ plane to obtain the average vector, which comprised the coordinates $n_{i_x}, n_{i_y}, n_{i_z}$. The methodology employed involved the removal of the empirical mean from the PCA data matrix to reduce variance. This was followed by performing SVD on the modified data matrix, as depicted in Eq. (2).

$$\min_{\mathbf{n}_i} \|\mathbf{A} \cdot \mathbf{n}_i\|_2 \quad (2)$$

$\mathbf{A} = \left[\mathbf{Q}_i^+ - \overline{\mathbf{q}}_i^+\right]$, where $\overline{\mathbf{q}}_i^+$ is the mean vector of $\mathbf{Q}_i^+$ obtained from $\frac{1}{n}\sum_{i=1}^{n} \mathbf{Q}_i^+$. The application of SVD to the modified matrix $\mathbf{A}$ was justified due to its ability to provide a singular solution, which in turn yields the normal vector perpendicular to the plane formed by matrix $\mathbf{A}$. Subsequently, the procedure of ascertaining the surface normal vector was furthered by establishing the unit vector employed as a point of reference for the vector's normal direction. The unit vector $\hat{\mathbf{n}}_i$ was obtained by normalizing the normal coordinates of the vector, where each normal vector $\mathbf{n}_i$ was divided by the magnitude of $\mathbf{n}_i$, denoted as $|\mathbf{n}_i|$, which was calculated as $\sqrt{n_{i_x}^2 + n_{i_y}^2 + n_{i_z}^2}$. During the surface normal vector estimation stage, a novel data structure was created to represent the coordinates of each point and its corresponding vector normal in the $\mathbf{R}$ matrix, as generated in Eq. (3).

$$\mathbf{R} = \begin{bmatrix} p_{i_{x_1}} & p_{i_{y_1}} & p_{i_{z_1}} & \hat{n}_{i_{x_1}} & \hat{n}_{i_{y_1}} & \hat{n}_{i_{z_1}} \\ p_{i_{x_2}} & p_{i_{y_2}} & p_{i_{z_2}} & \hat{n}_{i_{x_2}} & \hat{n}_{i_{y_2}} & \hat{n}_{i_{z_2}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{i_{x_n}} & p_{i_{y_n}} & p_{i_{z_n}} & \hat{n}_{i_{x_n}} & \hat{n}_{i_{y_n}} & \hat{n}_{i_{z_n}} \end{bmatrix} \quad (3)$$

Subsequently, the normal vector direction underwent feature extraction. Given that the vector was orthogonal to the surface, various types of normal vectors could be used as reference characteristics to exclude points in the planar plane. Among the various types of normal vectors, a point p_i with a unit vector coordinate axis value close to 1 or 0 represented a normal vector that was perpendicular to the y, z plane, x, z plane, or x, y plane.

Following an examination of the normal vector direction, segmentation was subsequently conducted using a histogram comprising 20 divisions within each 0.05 range increment from 0 to 1 for the x, y, and z axes. This analysis aimed to establish that comparable vector directions, which signified planar planes, were concentrated in proximity to either 0 or 1. Elimination was performed in the histogram region containing a substantial number of data points, as the presence of diverse coordinates suggests a planar surface.

3.3. Density-based Region Point Cluster Proposal

After eliminating the planar planes, cluster formations were generated by grouping clusters according to their density, which was determined by measuring the distance between points. In this stage, the implementation of the DBSCAN method was chosen based on its simplicity and efficiency[55)]. Additionally, the DBSCAN exhibited efficient computational performance when dealing with substantial datasets, as demonstrated by its ability to process such data within a relatively brief timeframe[49)]. The utilization of the DBSCAN algorithm enabled the grouping of 3D LiDAR point cloud data, which were segregated into distinct clouds by means of a planar surface, into clusters. The outcomes of this phase are depicted in matrix **S**, as illustrated in Eq. (4), where the first three columns in the matrix represent the point coordinates, and the fourth column represents the label l of the point cluster.

$$\mathbf{S} = \begin{bmatrix} p_{i_{x_1}} & p_{i_{y_1}} & p_{i_{z_1}} & l_1 \\ p_{i_{x_2}} & p_{i_{y_2}} & p_{i_{z_2}} & l_2 \\ \vdots & \vdots & \vdots & \vdots \\ p_{i_{x_n}} & p_{i_{y_n}} & p_{i_{z_n}} & l_n \end{bmatrix} \tag{4}$$

The following step was classifying the data regions or clusters that were formed through DBSCAN using a deep learning system based on direct point data. Before starting this process, the clusters were divided into two categories with two classes based on training and testing data distribution. In this study, the training category received a portion of 80% of the total data clusters, while the testing gets 20% of the total data clusters. Both categories were contained within clusters with human and non-human classes.

3.4. Direct Point Cluster Classification

The classification procedure in this investigation utilized a modified version of PointNet as classifier in this study. The augmentation process was employed to normalize the point cloud data input into the classifier, ensuring that each cluster contained an equal number of points. This normalization was necessary to maintain the integrity of the cluster structure when feeding the data into the classifier. The augmentation process was implemented during both the training and testing phases. The representation of the augmentation process for each cluster was depicted in Eq. (5),

$$\begin{cases} S_{i_{aug}} = S_i \overset{R}{\leftarrow} S_i\left(max\left(S_{i_t}\right)\right), \mathbf{S}_{i_n} > max\left(S_{i_t}\right) \\ S_{i_{aug}} = S_i \times \frac{max\left(S_{i_t}\right)}{\mathbf{S}_{i_n}} \overset{R}{\leftarrow} S_i\left(max\left(S_{i_t}\right)\right), \mathbf{S}_{i_n} < max\left(S_{i_t}\right) \\ S_{i_{aug}} = S_i, otherwise \end{cases} \tag{5}$$

where S_i was the the cluster proposed in the classifier, $\mathbf{S}_{i_n}$ was the number of points in the S_i cluster, and $max(S_{i_t})$ was the highest number of points in one of the clusters in matrix **S** categorized as training data. Therefore, $S_{i_{aug}}$ had the same number of points as the cluster with the highest number of points. Thus, the training and testing process could be executed based on the modified PointNet classifier.

Generally, PointNet utilizes the Rectified Linear Unit (ReLU) as the activation function within its neural network layer in its practical implementation. In this study, a minor alteration was implemented in PointNet by substituting the Rectified Linear Unit (ReLU) activation function with the Hyperbolic Tangent (Tanh) activation function. This approach was implemented in order to proactively address the potential challenges associated with the variability of negative calculated values and the intricate non-linear characteristics of 3D LiDAR point cloud data[56)].

To evaluate the efficacy of the classifier system, this research employed sparse categorical cross entropy as a loss function and accuracy metrics to assess the performance of the training model. Considering that the target labels were represented as integers rather than one-hot encoded vectors, this method was chosen to mitigate the substantial memory overhead that was typically incurred when representing one-hot encoded vectors. The aforementioned function demonstrated enhanced efficiency in terms of both memory utilization and computational processes. The primary aim was to reduce the disparity between the anticipated value and the desired value. This loss function evaluated the extent of misclassification induced by the model. Accuracy metrics were determined by assessing the agreement between the predicted class generated by the model and the actual target class. This evaluation was performed using sparse categorical accuracy. When the classification made by the model corresponds to the intended classification, it was considered that the data was correctly classified. Accuracy metrics provided a thorough assessment of the effectiveness of a model in performing classification tasks.

3.5. Distance Estimation

The detected object was finally generated within a bounding box, denoted as b_i, which encompassed points extracted from the maximum and minimum height, width, and length of the object. In order to facilitate the implementation of human monitoring systems, the inclusion of distance d as a crucial component was imperative for the successful operation of a tracking system. This study enhanced the detection procedure by employing distance measurements to ascertain the precise location of the individual under surveillance. The distance estimation pertained to the measurement of the distance between the centroid[57)] coordinates of the object's

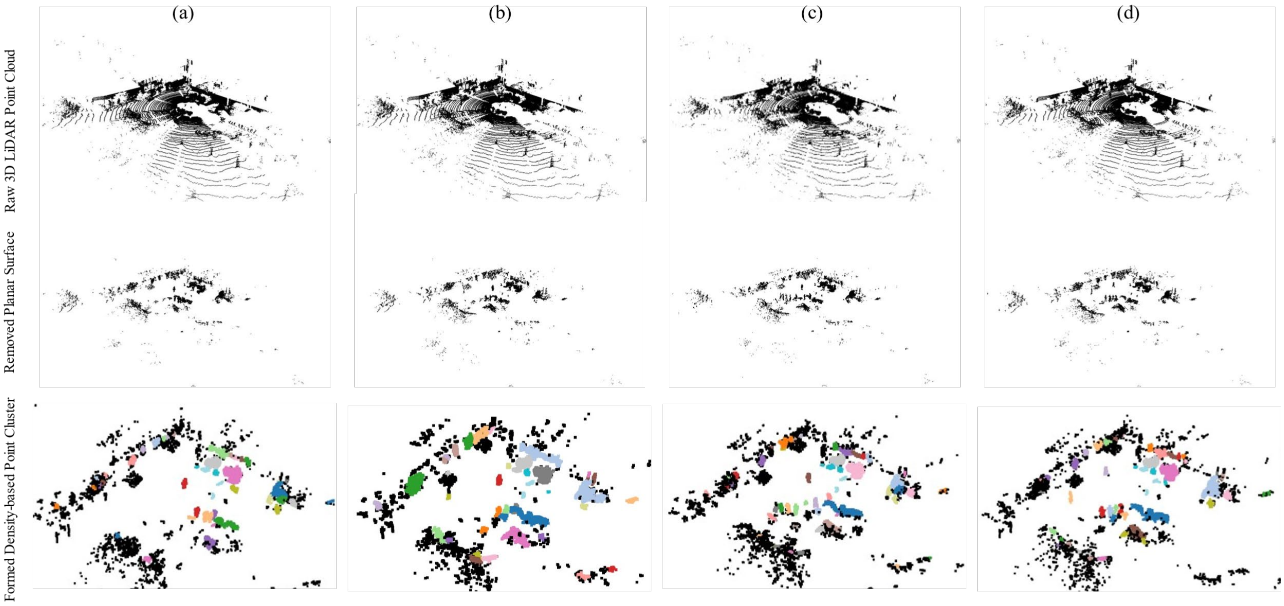


Fig. 4: Density-based cluster formation from the removed planar surface of raw 3D LiDAR point cloud KITTI dataset: (a) 2011_09_28_drive_0208; (b) 2011_09_28_drive_0216; (c) 2011_09_28_drive_0174; (d) 2011_09_28_drive_0199.

bounding box represented as $\bar{b}_i = \left[\bar{b}_{i_x},\ \bar{b}_{i_y},\ \bar{b}_{i_z}\right]^T$ resulting from $\frac{1}{n}\sum_{i=1}^{n} b_i$, and the 3D LiDAR position coordinates, represented in $p_l = \left[p_{l_x}, p_{l_y}, p_{l_z}\right]^T$, calculated using the 3D euclidean distance approach[58], with measurement unit in meter[20,59,60]. However, because the centroid height of the bounding box object varied, the p_{l_z} value also varied according to the $\bar{b}_{i_z}$ value. Consequently, the distance estimation equation in this study was as presented on Eq. (6).

$$d = \sqrt{\left(\bar{b}_{i_x} - p_{l_x}\right)^2 + \left(\bar{b}_{i_y} - p_{l_y}\right)^2 + \left(\bar{b}_{i_z} - p_{l_z}\right)^2} \quad (6)$$

4. Results and Discussion

The four datasets used in this study are obtained from the KITTI raw 3D LiDAR point cloud dataset 16). Those datasets include 2011_09_28_drive_0208 and 2011_09_28_drive_0216, which capture the scenario of a human walking towards the LiDAR point. In the similar characteristic with different scenario, dataset 2011_09_28_drive_0174 and 2011_09_28_drive_0199 have been used, which capture the scenario of a human walking away from the LiDAR point. The dataset labeled as ”2011_09_28_drive_0208” is utilized for both training the model and conducting testing. Conversely, the remaining three datasets are exclusively employed for inference purposes.

In order to facilitate the training process and subsequent inference testing of the methodology employed in this study, those datasets undergo an initial planar surface removal procedure to eliminate connections between objects linked by a 2D x, y plane. In this phase, the SVD technique is utilized to analyze the PCA plane that is generated using kdTree-NN. Subsequently, an elimination process is performed based on the analysis of the histogram of the direction of the normal vectors. Following the successful completion of the planar surface removal, the subsequent phase involves the construction of clusters that can be seen in Fig. 4. This study employs the DBSCAN method approach to form density-based clusters. In contrast, the RANSAC approach was not used due to its limited effectiveness in processing 3D LiDAR point cloud data characterized by intricate structures, such as the presence of overlapping or irregular objects. The DBSCAN algorithm operates by partitioning data points into clusters based on their proximity within a specified distance threshold. The cluster construction process yields multiple groupings of point clouds that share the same cluster label. Therefore, the points sharing the same label are assigned to a common cluster, which subsequently serves as the input data for the classifier.

The clusters obtained from the DBSCAN-based cluster construction are incorporated into the training and inference procedures. In this study, the dataset selected to construct a modified PointNet-based classifier model is the 2011_09_28_drive_0208 dataset. As previously stated, it is necessary for PointNet classifier to get input data that consist of point cloud with an equal number of points. Subsequently, the augmentation procedure is executed. The study reveals that the maximum value of S_{it} is 818. Consequently, all input points are standardized to a uniform size of 818×3, encompassing human and non-human classes. Once the augmentation process is finalized, the training process commences, consisting of 100 iterations.

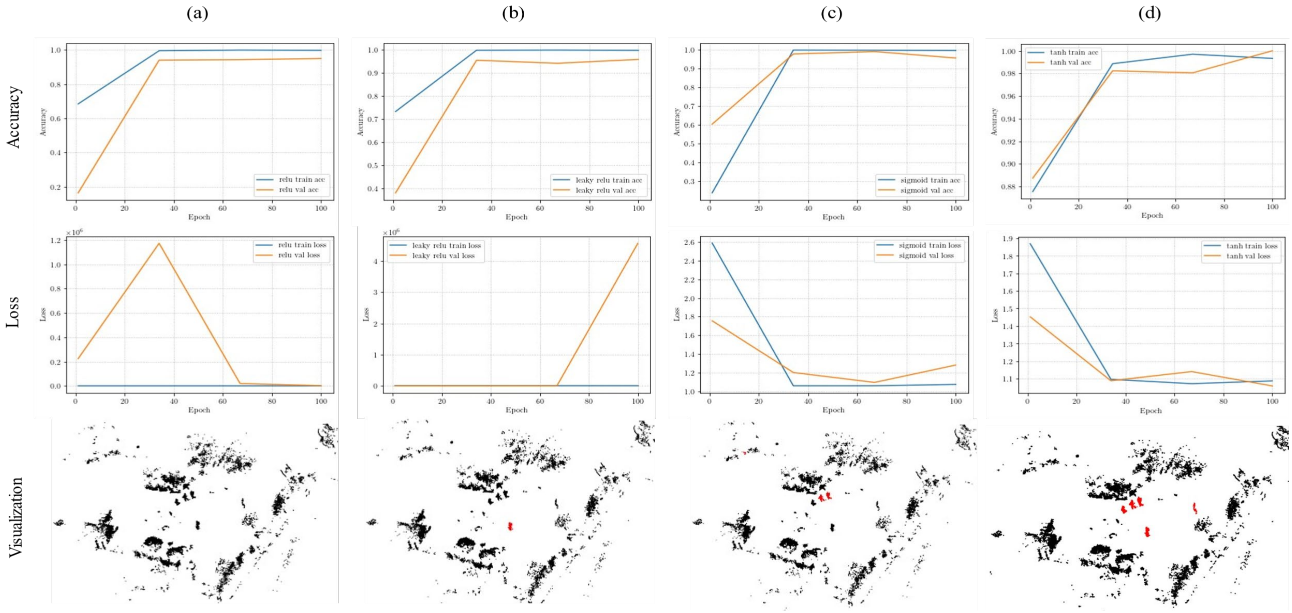


Fig. 5: Training and validation sampling result between default and modified classifier: (a) Default PointNet classifier with ReLU activation function; (b) PointNet classifier with Leaky ReLU activation function; (c) PointNet classifier with Sigmoid activation function; (d) (Ours) Modified PointNet with Tanh activation function.

The implementation of PointNet yields a substantial validation error, rendering it incapable of accurately identifying objects across all classes within the training and testing datasets. Consequently, a thorough investigation is undertaken, revealing that the utilization of the ReLU activation function proved ineffective in addressing the specific data employed within this research. The negative output value observed in the layer is attributed to the intricate nature of the input shape of the 3D LiDAR point cloud. This complexity arises from the presence of blank spots on various objects, leading to variations in the input shape. Consequently, the PointNet network is subjected to modification through the utilization of alternative activation functions. This study examines the training process using four frequently employed activation functions: ReLU, Leaky ReLU, Sigmoid, and Tanh. The findings indicate that Tanh exhibited superior performance in replacing ReLU within the neural network layer. The sampling outcomes of the classifier's training, following the adjustment of the activation function, are depicted in Fig. 5.

Figure. 5 illustrates the comparative accuracy achieved by four activation functions. The accuracy values obtained for ReLU, Leaky ReLU, Sigmoid, and Tanh are 94.98%, 95.91%, 95.72%, and 98.79%, respectively. Nevertheless, three out of the four activation functions exhibited overfitting. This is evidenced by the unfavorable outcomes observed in the loss results. When implementing the ReLU activation function, it is observed that although the output values may seem close to zero, the validation losses during the training process can reach a maximum loss of 4.26×10^{10} and 1.40×10^{3} at the final iteration. During the implementation of Leaky ReLU, it was observed that the utilization of this activation function led to a validation loss during the training process, reaching a maximum loss of 5.53×10^{6} and concluding with a loss of 4.56×10^{6} at the end of the iteration. The utilization of the Sigmoid function in the experiment revealed that during the training process, the validation loss initially decreased to a minimum value of 1.05. However, it subsequently increased to a final loss value of 1.28 by the end of the iteration. In the context of utilizing the Tanh function, it was observed that employing Tanh led to a validation loss during the training phase, reaching a minimum loss of 1.05 and steadily decreasing until the completion of the iteration.

This study modifies the default PointNet architecture to process 3D point clouds. Some of relevant studies are also inspired by PointNet architecture but none of them employs PointNet architecture directly especially when the reference is humans. In order to study the efficiency, especially in terms of the speed and accuracy, of this method and other state-of-the-art approaches, the researcher compared the result of the studies which employ KITTI dataset in human object detection. The F-PointNets[42] study results in 5.9 FPS with accuracy 50.53%. IPOD[61] method results in 5 FPS with accuracy 55.07%. Both of these methods incorporate two types of data modality namely 3D LiDAR point cloud and RGB image. Consequently, the computational cost is relatively high due to the low FPS. Meanwhile, another method named 3DSSD[62] employs only one modality which is 3D LiDAR point cloud achieves significant speed up to 25 FPS. However, as this method uses single-stage detector without object proposal stage, the accuracy percentage is only 54.64%. To achieve a better result, this current study applies region proposal on modified PointNet classifier so that the accuracy has been successfully improved up to

98.79% and the speed has also been improved up to 27.03 FPS. Figure. 6 presents the comparison of the results.

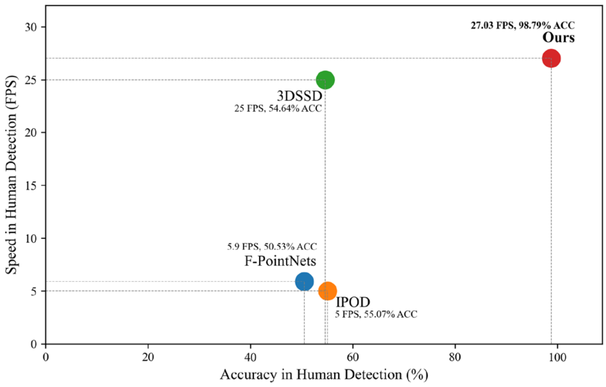


Fig. 6: PointNet inspired state-of-the-art performance comparison.

Following the steps of training and testing, the next step entails carrying out the inference process. The inference process is carried out in order to simultaneously evaluate the dependability of the framework in terms of calculating the distance that exists between a human and a LiDAR device, with a particular emphasis placed on the walking behavior of the human. The datasets utilized in this step include 2011_09_28_ drive_0216, 2011_09_28_ drive_01 74, and 2011_09_28_drive_0199. Additionally, an inference is performed on the training dataset 2011_09_28_drive_0208 for comparative purposes. The reason for this is that, during the evaluation of the testing data from the 2011_09_28_drive_0208 dataset, the distance estimation process yielded successful results. As a result, the system was able to derive distance estimates for all detected human objects. The outcomes of the distance estimation for all identified human objects can be seen in Fig. 7.

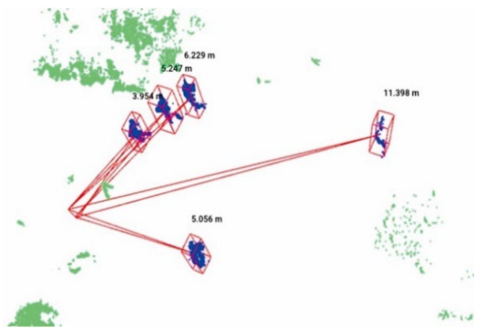


Fig. 7: 3D LiDAR human point cloud tracked with measured distance from LiDAR position.

The process of measuring distance involves initially determining the midpoint of the bounding box surrounding the human object. Subsequently, the measurement of the distance between the two points is conducted using the 3D Euclidean distance, thereby obtaining an approximation of the distance. Figure. 7 displays the outcomes of the process of determining the approximate distance between the human object and the LiDAR.

The effectiveness of the human object tracking process is evident in Fig. 8, where the process is facilitated by consistent data updates derived from the estimated distance between the human object and the LiDAR sensor. This is demonstrated in scenarios where the human subject is in motion, either approaching or moving away from the LiDAR sensor. In the context of the human walking process in relation to LiDAR, it is evident that the estimated distance decreases progressively from the initial point to the final point of the scene. In contrast to the situation where humans are moving away from LiDAR, it is observed that the calculated distance exhibits a progressively greater magnitude from the initial to the final portion of the scene.

The dataset 2011_09_28_drive_0208 contains distance fluctuations to track a person near the LiDAR sensor. In the final frame, the individual's position changes from 11.178 m to 4.131 m. In 2011_09_28_drive_0216, the LiDAR sensor accurately tracks a person walking near it. The distance between the person and the LiDAR sensor decreases from 11.441 m to 3.753 m. 2011_09_28_drive_ 0174 was used to be reference data to analyze distance fluctuations. These data showed a systematic tracking of the individual's displacement relative to the LiDAR sensor. The last frame shows a distance increase from 4.052 m to 11.603 m. A distance change was recorded in the dataset 2011_09_28_drive_0199, showing systematic tracking of human's movement away from the LiDAR device. The data covers all frames, showing a distance increment from 6.857 m to 11.401 m.

In general, the performance of the detection and tracking system, which incorporates a modified PointNet classifier for object distance estimation in the proposed framework, is satisfactory. The system effectively identifies and retrieves the distance of human objects solely using a single data type, specifically the raw 3D LiDAR point cloud. Therefore, the system can be suggested as a means of supporting the human monitoring work in future research endeavors.

5. Conclusion

The study found that the region cluster proposal approach for modified PointNet classifier-based direct point processing effectively facilitates detection and tracking in this human monitoring framework. The modification yielded an improved classification accuracy percentage during the detection process, achieving an accuracy of 98.79% compared to the default PointNet's accuracy of 94.98%. Conversely, the tracking procedure was effectively executed to observe the spatial proximity of objects relative to the LiDAR location in the context of individuals approaching and moving away from the LiDAR. It is observed that during the process of walking in proximity to or moving away from the LiDAR position,

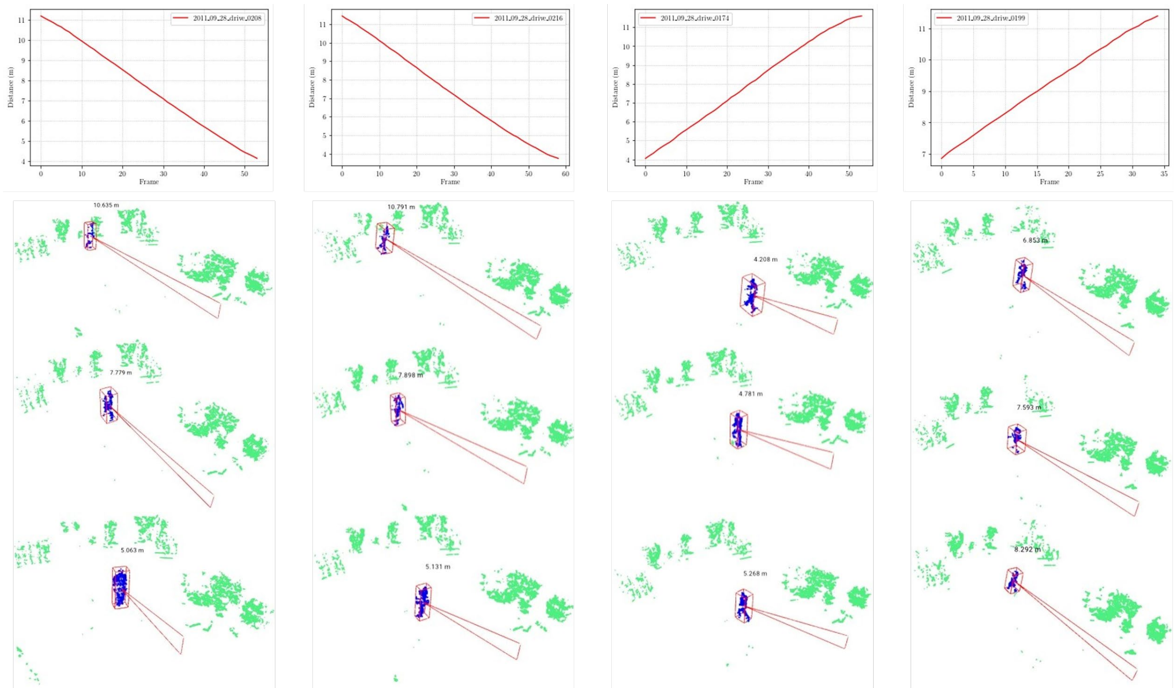


Fig. 8: Inference human distance monitoring results on different dataset scenario: (a) Human walking towards LiDAR (2011_09_28_drive_0208); (b) Human walking towards LiDAR (2011_09_28_drive_0216); (c) Human walking away from LiDAR (2011_09_28_drive_0174); (d) Human walking away from LiDAR (2011_09_28_drive_0199).

all objects are detected and tracked including their respective distances. Hence, the suggested framework presents a viable option for research pertaining to human monitoring work, as it relies solely on a singular form of unprocessed 3D LiDAR point cloud data while simultaneously preserving the privacy of the individual under surveillance.

Acknowledgements

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, the Republic of Indonesia, and the University Center of Excellence on Artificial Intelligence for Healthcare and Society (UCE AIHeS) Institut Teknologi Sepuluh Nopember, in collaboration with Graduate School of Science and Technology Kumamoto University for fully supporting this work.

References

1) S. Maity, D. Das, and S. Sen, "Wearable health monitoring using capacitive voltage-mode Human Body Communication," in: 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, Seogwipo, 2017: pp. 1–4. doi:10.1109/EMBC.2017.8036748.
2) M. Sharma, and M. Soni, "A musculoskeletal finite element study of a unique and customised jaw joint prosthesis for the asian populace," *Evergreen*, **7** (*3*) 351–358 (2020). doi:10.5109/4068615.
3) S. Davila-Montero, J.A. Dana-Le, G. Bente, A.T. Hall, and A.J. Mason, "Review and challenges of technologies for real-time human behavior monitoring," *IEEE Trans. Biomed. Circuits Syst.*, **15** (*1*) 2–28 (2021). doi:10.1109/TBCAS.2021.3060617.
4) A. Habib, K M Ariful Kabir, and J. Tanimoto, "'Do humans play according to the game theory when facing the social dilemma situation?' a survey study.," *Evergreen*, **7** (*1*) 7–14 (2020). doi:10.5109/2740936.
5) R. Maurya, T.H. Teo, S.H. Chua, H.-C. Chow, and I.-C. Wey, "Complex Human Activities Recognition Based on High Performance 1D CNN Model," in: 2022 IEEE 15th International Symposium on Embedded Multicore/Many-Core Systems-on-Chip (MCSoC), IEEE, Penang, Malaysia, 2022: pp. 330–336. doi:10.1109/MCSoC57363.2022.00059.
6) P. Panwar, P. Roshan, R. Singh, M. Rai, Asha Rani Mishra, and Sansar Singh Chauhan, "DDNet- a deep learning approach to detect driver distraction and drowsiness," *Evergreen*, **9** (*3*) 881–892 (2022). doi:10.5109/4843120.
7) J. Manjarres, V. Russo, J. Penaranda, and M. Pardo, "Human Activity and Heart Rate Monitoring System in a Mobile Platform," in: 2018 Congreso Internacional de Innovación y Tendencias En Ingeniería (CONIITI), IEEE, Bogota, 2018: pp. 1–6. doi:10.1109/CONIITI.2018.8587094.

8) S. Wahyudi, M. Ridwan F, and Putu Hadi Setyarini, "Effects of metabolic heat on the temperature distribution of human hands affected by sarcoma tumors given interstitial hyperthermia therapy," *Evergreen*, **9** (*2*) 262–268 (2022). doi:10.5109/4793633.
9) U. Gurnani, Sanjay Kumar Singh, Manoj Kumar Sain, and M. L. Meena, "Musculoskeletal health problems and their association with risk factors among manual dairy farm workers," *Evergreen*, **9** (*4*) 950–961 (2022). doi:10.5109/6622881.
10) Q. Cai, "Human Behavior Recognition Algorithm Based on HOG Feature and SVM Classifier," in: 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), IEEE, Beijing, China, 2019: pp. 233–236. doi:10.1109/ICSESS47205.2019.9040826.
11) F. Yulia, S. Marsya, Y. Bobby, Nasruddin, and A Zulys, "Design and preparation of succinic acid-based metal-organic frameworks for co_2 adsorption technology," *Evergreen*, **7** (*4*) 549–554 (2020). doi:10.5109/4150475.
12) G. Bassani, A. Filippeschi, and C.A. Avizzano, "A dataset of human motion and muscular activities in manual material handling tasks for biomechanical and ergonomic analyses," *IEEE Sensors J.*, **21** (*21*) 24731–24739 (2021). doi:10.1109/JSEN.2021.3113123.
13) K.K. Verma, B.M. Singh, H.L. Mandoria, and P. Chauhan, "Two-stage human activity recognition using 2d-convnet," *IJIMAI*, **6** (*2*) 11 (2020). doi:10.9781/ijimai.2020.04.002.
14) M.N. Bhuiyan, M.M. Billah, F. Bhuiyan, M.A.R. Bhuiyan, N. Hasan, M.M. Rahman, M.S. Miah, M. Alibakhshikenari, F. Arpanaei, F. Falcone, and M. Niu, "Design and implementation of a feasible model for the iot based ubiquitous healthcare monitoring system for rural and urban areas," *IEEE Access*, **10** 91984–91997 (2022). doi:10.1109/ACCESS.2022.3202551.
15) Lalit N. Patil and Hrishikesh P. Khairnar, "Investigation of human safety based on pedestrian perceptions associated to silent nature of electric vehicle," *Evergreen*, **8** (*2*) 280–289 (2021). doi:10.5109/4480704.
16) A. Shokouhmand, S. Eckstrom, B. Gholami, and N. Tavassolian, "Camera-augmented non-contact vital sign monitoring in real time," *IEEE Sensors J.*, **22** (*12*) 11965–11978 (2022). doi:10.1109/JSEN.2022.3172559.
17) W. Wijayanti, H. Miawarni, N.E. Budiyanta, D.A. Rivai, and M.H. Purnomo, "Fall detection system for elderly activity daily living using modified 2d lidar as a low-cost alternative to 3d lidar based on the slice feature approach," *International Journal on Electrical Engineering and Informatics*, **15** (*2*) 291–304 (2023). doi:10.15676/ijeei.2023.15.2.8.
18) A. Naser, A. Lotfi, M.D. Mwanje, and J. Zhong, "Privacy-preserving, thermal vision with human in the loop fall detection alert system," *IEEE Trans. Human-Mach. Syst.*, **53** (*1*) 164–175 (2023). doi:10.1109/THMS.2022.3203021.
19) J. Ahn, K. Nakashima, K. Yoshino, Y. Iwashita, and R. Kurazume, "2V-Gait: Gait Recognition using 3D LiDAR Robust to Changes in Walking Direction and Measurement Distance," in: 2022 IEEE/SICE International Symposium on System Integration (SII), IEEE, Narvik, Norway, 2022: pp. 602–607. doi:10.1109/SII52469.2022.9708899.
20) A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: the kitti dataset," *The International Journal of Robotics Research*, **32** (*11*) 1231–1237 (2013). doi:10.1177/0278364913491297.
21) Y. Zhou, and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Salt Lake City, UT, USA, 2018: pp. 4490–4499. doi:10.1109/CVPR.2018.00472.
22) Y. Zhou, P. Sun, Y. Zhang, D. Anguelov, J. Gao, T. Ouyang, J. Guo, J. Ngiam, and V. Vasudevan, "End-to-end multi-view fusion for 3d object detection in lidar point clouds," (2019). doi:10.48550/ARXIV.1910.06528.
23) M. Alsfasser, J. Siegemund, J. Kurian, and A. Kummert, "Exploiting polar grid structure and object shadows for fast object detection in point clouds," in: W. Osten, D.P. Nikolaev (Eds.), Twelfth International Conference on Machine Vision (ICMV 2019), SPIE, Amsterdam, Netherlands, 2020: p. 27. doi:10.1117/12.2557043.
24) K. Zhang, S. Qiao, and K. Gao, "A new point cloud reconstruction algorithm based-on geometrical features," in: 2015 7th International Conference on Modelling, Identification and Control (ICMIC), IEEE, Sousse, Tunisia, 2015: pp. 1–6. doi:10.1109/ICMIC.2015.7409387.
25) Zhirong Wu, S. Song, A. Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Boston, MA, USA, 2015: pp. 1912–1920. doi:10.1109/CVPR.2015.7298801.
26) 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: TENCON 2021 - 2021 IEEE Region 10 Conference (TENCON), IEEE, Auckland, New Zealand, 2021: pp. 632–636. doi:10.1109/TENCON54134.2021.9707317.
27) 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., **52** (*10*) 10027–10040 (2022). doi:10.1109/TCYB.2021.3085489.

28) H. Liu, Q. Hu, Y. Ma, K. Xie, and Y. Guo, "DeepPCT: single object tracking in dynamic point cloud sequences," *IEEE Trans. Instrum. Meas.*, **72** 1–12 (2023). doi:10.1109/TIM.2022.3232092.
29) Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3d point clouds: a survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, **43** (*12*) 4338–4364 (2021). doi:10.1109/TPAMI.2020.3005434.
30) 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*, **7** 68892–68904 (2019). doi:10.1109/ACCESS.2019.2918862.
31) D. Maturana, and S. Scherer, "VoxNet: A 3D Convolutional Neural Network for real-time object recognition," in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Hamburg, Germany, 2015: pp. 922–928. doi:10.1109/IROS.2015.7353481.
32) L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese, "SEGCloud: Semantic Segmentation of 3D Point Clouds," in: 2017 International Conference on 3D Vision (3DV), IEEE, Qingdao, 2017: pp. 537–547. doi:10.1109/3DV.2017.00067.
33) R.Q. Charles, H. Su, M. Kaichun, and L.J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Honolulu, HI, 2017: pp. 77–85. doi:10.1109/CVPR.2017.16.
34) H. Seo, and S. Joo, "Influence of Preprocessing and Augmentation on 3D Point Cloud Classification Based on a Deep Neural Network: PointNet," in: 2020 20th International Conference on Control, Automation and Systems (ICCAS), IEEE, Busan, Korea (South), 2020: pp. 895–899. doi:10.23919/ICCAS50221.2020.9268197.
35) Z. Zhongyang, C. Yinglei, S. Xiaosong, Q. Xianxiang, and S. Li, "Classification of LiDAR Point Cloud based on Multiscale Features and PointNet," in: 2018 Eighth International Conference on Image Processing Theory, Tools and Applications (IPTA), IEEE, Xi'an, 2018: pp. 1–7. doi:10.1109/IPTA.2018.8608120.
36) 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: 2023 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), IEEE, Kuala Lumpur, Malaysia, 2023: pp. 1–6. doi:10.1109/I2MTC53148.2023.10175928.
37) B. Yang, W. Luo, and R. Urtasun, "PIXOR: Real-time 3D Object Detection from Point Clouds," in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Salt Lake City, UT, USA, 2018: pp. 7652–7660. doi:10.1109/CVPR.2018.00798.
38) J. Beltran, C. Guindel, F.M. Moreno, D. Cruzado, F. Garcia, and A. De La Escalera, "BirdNet: A 3D Object Detection Framework from LiDAR Information," in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, Maui, HI, 2018: pp. 3517–3523. doi:10.1109/ITSC.2018.8569311.
39) A. Barrera, C. Guindel, J. Beltran, and F. Garcia, "BirdNet+: End-to-End 3D Object Detection in LiDAR Bird's Eye View," in: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), IEEE, Rhodes, Greece, 2020: pp. 1–6. doi:10.1109/ITSC45102.2020.9294293.
40) A. Barrera, J. Beltran, C. Guindel, J.A. Iglesias, and F. Garcia, "BirdNet+: two-stage 3d object detection in lidar through a sparsity-invariant bird's eye view," *IEEE Access*, **9** 160299–160316 (2021). doi:10.1109/ACCESS.2021.3131389.
41) S. Mohapatra, S. Yogamani, H. Gotzig, S. Milz, and P. Mader, "BEVDetNet: Bird's Eye View LiDAR Point Cloud based Real-time 3D Object Detection for Autonomous Driving," in: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), IEEE, Indianapolis, IN, USA, 2021: pp. 2809–2815. doi:10.1109/ITSC48978.2021.9564490.
42) C.R. Qi, W. Liu, C. Wu, H. Su, and L.J. Guibas, "Frustum PointNets for 3D Object Detection from RGB-D Data," in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Salt Lake City, UT, USA, 2018: pp. 918–927. doi:10.1109/CVPR.2018.00102.
43) P. Cao, H. Chen, Y. Zhang, and G. Wang, "Multi-View Frustum Pointnet for Object Detection in Autonomous Driving," in: 2019 IEEE International Conference on Image Processing (ICIP), IEEE, Taipei, Taiwan, 2019: pp. 3896–3899. doi:10.1109/ICIP.2019.8803572.
44) G.H. Golub, and C. Reinsch, "Singular value decomposition and least squares solutions," *Numerische Mathematik*, **14** (*5*) 403–420 (1970). doi:10.1007/BF02163027.
45) H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *SIGGRAPH Comput. Graph.*, **26** (*2*) 71–78 (1992). doi:10.1145/142920.134011.
46) A. Kumar, Arindam Kumar Chanda, and S. Angra, "Optimization of stiffness properties of composite sandwich using hybrid taguchi-gra-pca," *Evergreen*, **8** (*2*) 310–317 (2021). doi:10.5109/4480708.
47) J.L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, **18** (*9*) 509–517 (1975). doi:10.1145/361002.361007.

48) 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*, **24** (*6*) 381–395 (1981). doi:10.1145/358669.358692.
49) M. Ester, H.-P. Kriegel, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in: KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, AAAI Press, Portland Oregon, 1996: pp. 226–231. doi:https://dl.acm.org/doi/10.5555/3001460.3001507.
50) T.-H. Wu, C. Lian, S. Lee, M. Pastewait, C. Piers, J. Liu, F. Wang, L. Wang, C.-Y. Chiu, W. Wang, C. Jackson, W.-L. Chao, D. Shen, and C.-C. Ko, "Two-stage mesh deep learning for automated tooth segmentation and landmark localization on 3d intraoral scans," *IEEE Trans. Med. Imaging*, **41** (*11*) 3158–3166 (2022). doi:10.1109/TMI.2022.3180343.
51) H. Matsumura, and C. Premachandra, "Deep-learning-based stair detection using 3d point cloud data for preventing walking accidents of the visually impaired," *IEEE Access*, **10** 56249–56255 (2022). doi:10.1109/ACCESS.2022.3178154.
52) J. Kossaifi, A. Bulat, G. Tzimiropoulos, and M. Pantic, "T-Net: Parametrizing Fully Convolutional Nets With a Single High-Order Tensor," in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Long Beach, CA, USA, 2019: pp. 7814–7823. doi:10.1109/CVPR.2019.00801.
53) Y. Lyu, X. Huang, and Z. Zhang, "Revisiting 2d convolutional neural networks for graph-based applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, **45** (*6*) 6909–6922 (2023). doi:10.1109/TPAMI.2021.3083614.
54) C. Ding, L. Zhang, H. Chen, H. Hong, X. Zhu, and F. Fioranelli, "Sparsity-based human activity recognition with pointnet using a portable fmcw radar," *IEEE Internet Things J.*, **10** (*11*) 10024–10037 (2023). doi:10.1109/JIOT.2023.3235808.
55) D. Deng, "DBSCAN Clustering Algorithm Based on Density," in: 2020 7th International Forum on Electrical Engineering and Automation (IFEEA), IEEE, Hefei, China, 2020: pp. 949–953. doi:10.1109/IFEEA51475.2020.00199.
56) R. Wang, J. Peethambaran, and D. Chen, "LiDAR point clouds to 3-d urban models: a review," *IEEE J. Sel. Top. Appl. Earth Observations Remote Sensing*, **11** (*2*) 606–627 (2018). doi:10.1109/JSTARS.2017.2781132.
57) I. Surjandari, R. Rindrasari, and A. Dhini, "Evaluation of efficiency in logistics company: an analysis of last-mile delivery," *Evergreen*, **10** (*2*) 649–657 (2023). doi:10.5109/6792811.
58) L. Wen, L. He, and Z. Gao, "Research on 3d point cloud de-distortion algorithm and its application on euclidean clustering," *IEEE Access*, **7** 86041–86053 (2019). doi:10.1109/ACCESS.2019.2926424.
59) G. Engels, N. Aranjuelo, I. Arganda-Carreras, M. Nieto, and O. Otaegui, "3D Object Detection from LiDAR Data using Distance Dependent Feature Extraction:," in: Proceedings of the 6th International Conference on Vehicle Technology and Intelligent Transport Systems, SCITEPRESS - Science and Technology Publications, Prague, Czech Republic, 2020: pp. 289–300. doi:10.5220/0009330402890300.
60) F. Nezhadalinaei, L. Zhang, M. Mahdizadeh, and F. Jamshidi, "Motion Object Detection and Tracking Optimization in Autonomous Vehicles in Specific Range with Optimized Deep Neural Network," in: 2021 7th International Conference on Web Research (ICWR), IEEE, Tehran, Iran, 2021: pp. 53–63. doi:10.1109/ICWR51868.2021.9443120.
61) Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "IPOD: intensive point-based object detector for point cloud," (2018). doi:10.48550/ARXIV.1812.05276.
62) Z. Yang, Y. Sun, S. Liu, and J. Jia, "3DSSD: Point-Based 3D Single Stage Object Detector," in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Seattle, WA, USA, 2020: pp. 11037–11045. doi:10.1109/CVPR42600.2020.01105.