ISSN : 1226-7244 (Print)
ISSN : 2288-243X (Online)
논문번호 23-04-11
72
j.inst.Korean.electr.electron.eng.Vol.27,No.4,439~449,December 2023
https : //doi.org/10.7471/ikeee.2023.27.4.439

Indoor Environment Drone Detection through DBSCAN and Deep Learning

Ha Tran Thi*, Hien Pham The*, Yun-Seok Mun*, Ic-Pyo Hong*★

Abstract

In an era marked by the increasing use of drones and the growing demand for indoor surveillance, the development of a robust application for detecting and tracking both drones and humans within indoor spaces becomes imperative. This study presents an innovative application that uses FMCW radar to detect human and drone motions from the cloud point. At the outset, the DBSCAN (Density-based Spatial Clustering of Applications with Noise) algorithm is utilized to categorize cloud points into distinct groups, each representing the objects present in the tracking area. Notably, this algorithm demonstrates remarkable efficiency, particularly in clustering drone point clouds, achieving an impressive accuracy of up to 92.8%. Subsequently, the clusters are discerned and classified into either humans or drones by employing a deep learning model. A trio of models, including Deep Neural Network (DNN), Residual Network (ResNet), and Long Short-Term Memory (LSTM), are applied, and the outcomes reveal that the ResNet model achieves the highest accuracy. It attains an impressive 98.62% accuracy for identifying drone clusters and a noteworthy 96.75% accuracy for human clusters.

Key words : DBSCAN, Deep learning, FMCW radar, Object detection, UAV.

Ⅰ. Introduction

Drones have transcended their early role as aerial cameras and have assumed multifaceted roles across industries. They now play pivotal roles in agriculture, capturing data for precision.

Environmental scientists and researchers have turned to drones to collect data for ecological studies and environmental monitoring. In [2], Authors use drones and air quality sensors in environmental monitoring of air pollutant emissions. Otherwise, Drones offer an efficient and cost-effective means of inspecting critical infrastructure, including bridges, power lines, and communication towers. These applications contribute to the timely detection of maintenance issues and the prevention of potential hazards, enhancing overall infrastructure safety [3]. Nonetheless, due to the rapid growth of the drone manufacturing industry, these unmanned aerial vehicles have become quite cost-effective for consumers, readily available for purchase, and user-friendly. Consequently, this accessibility has led to an increase in individuals utilizing drones for nefarious purposes, such as illegal surveillance or data collection. In response to this challenge, various studies have been

* Dept. of Smart Information and Technology Engineering, Kongju National University
★ Corresponding author
E-mail : iphong@kongju.ac.kr, Tel : +84-41-521-9199
※ Acknowledgment
This work was supported in part by the Basic Science Research Program under Grant 2020R1I1A3057142, and in part by the Underground City of the Future Program funded by the Ministry of Science and ICT.
Manuscript received Nov. 1. 2023; revised Nov, 25, 2023; accepted Dec, 5, 2023.

conducted, suggesting methods to detect and differentiate drones from other aerial entities, like birds, enabling the timely alerting of authorities when drones encroach upon restricted or private spaces. Nonetheless, due to the rapid growth of the drone manufacturing industry, these unmanned aerial vehicles have become quite cost-effective for consumers, readily available for purchase, and user-friendly. Consequently, this accessibility has led to an increase in individuals utilizing drones for nefarious purposes, such as illegal surveillance or data collection. In response to this challenge, various studies have been conducted, suggesting methods to detect and differentiate drones from other aerial entities, like birds, enabling the timely alerting of authorities when drones encroach upon restricted or private spaces [4], [5], [6].

In indoor environment, drones also have found various applications such as warehouse inventory management [7], [8] or security monitoring within shopping malls, airports. However, drones primarily depend on GPS (Global Positioning System) for their location, and GPS tends to perform inadequately in indoor settings. Consequently, the necessity for a system capable of detecting and tracking drones becomes critical to ensure the smooth operation of drones. Otherwise, accurately distinguishing between drones and humans within indoor settings carries significant importance across multiple areas. Primarily, this differentiation plays a vital role in ensuring safety, averting potential collisions, and protecting individuals who share indoor spaces with drones. Precise identification enables the implementation of safety measures, reducing the associated risks of drone operations and safeguarding the physical well-being of occupants. Additionally, the accurate classification of drones and humans indoors not only improves safety and security but also ensures adherence to regulations, fostering a conducive and well-organized indoor environment. As a result, this paper is dedicated to the development of an application aimed at addressing security, privacy, and safety concerns within indoor environments. Our application is designed to identify and monitor both humans and drones, and it relies on radar technology, specifically the Frequency-Modulated Continuous-Wave (FMCW) Radar. FMCW radar is chosen for its superior performance in various conditions, including situations with fog and limited visibility. Moreover, FMCW radar delivers exceptional precision in measuring distances, facilitating accurate tracking over extended ranges. An additional advantage is that FMCW radar operates continuously, ensuring uninterrupted surveillance, which is particularly well-suited for security and monitoring applications. This article introduces a method for processing point cloud data obtained from radar. It involves clustering the point clouds to identify the quantity of objects within them and then utilizing deep learning algorithms to predict whether a cluster corresponds to a human or a drone, ultimately providing precise location information for the detected objects. For the clustering of point clouds, we employ the DBSCAN algorithm, which relies on point density to delineate clusters. In terms of cluster detection, three deep learning models, specifically DNN, ResNet, and LSTM, are employed to determine which model achieves the highest accuracy when evaluated against our dataset.

The structure of this paper is as follows. In section II, we will delve into the theory and methodology associated with the algorithms utilized. Section III is dedicated to detailing the experimental setup and the real-world outcomes of our application. Lastly, in section IV, we will wrap up with the conclusion and future work.

II. Methodology and Relating background

1. Methodology

As illustrated in Figure 1, our research methodology involves two main phases: the Training and testing phase and the Actual Tracking phase. In

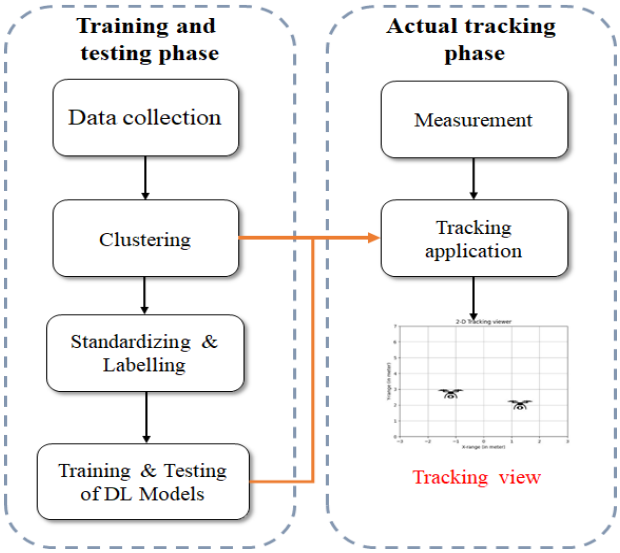


Fig. 1. Flowchart of methodology.

the initial Training and Testing of the Model phase, the process starts with data acquisition through an FMCW radar, which provides point cloud information. We then apply DBSCAN techniques to cluster the data within the point cloud. After this preprocessing, the clusters are categorized into one of three labels: 0 for drones, 1 for humans, and -1 for noise. The data is then standardized to create appropriate features, which are used as input data for the subsequent training and testing of Deep Learning models. We employ three different Deep Learning models to determine the identification of humans or drones within the clusters, comparing their accuracy to select the most suitable model for use in the Actual Tracking phase.

In the Actual Tracking phase, we continue to utilize radar data acquisition, similar to the initial step in the previous phase. Subsequently, the data concerning the points within each static frame is input into the tracking application. Within the tracking application, the point clouds are subjected to DBSCAN clustering to segment them into individual objects within the tracking area. Following this, the cluster data is standardized into input features, enabling the deep learning model to categorize the clusters. The ultimate result generated by the tracking application is a two-dimensional tracking viewer, presenting about the object types within the tracking area and their respective coordinates, as depicted in Figure 1.

2. DBSCAN algorithm

DBSCAN [9] focuses on identifying dense regions in the data space and separating them from sparse areas. Unlike traditional clustering algorithms, DBSCAN can automatically determine the number of clusters in the data. This property makes it highly suitable for real-world datasets where the number of clusters is unknown or variable.

According to the density connectivity concept, which underpins DBSCAN, data points are deemed to be a part of the same cluster if the are close enough to one another and have a local density of data points that is higher than a predetermined threshold. The following are the main ideas behind the DBSCAN algorithm:

Epsilon (ϵ) Neighborhood: DBSCAN classifies a neighborhood as a collection of data points that are all within a certain ϵ *distance* from each other. Every point that is situated inside the ϵ *-neighborhood* of another point is deemed capable of becoming a member of the cluster.

Core Points: A data point is deemed core if its ϵ*-neighborhood* contains at least data points *(minPts)*, including itself. Core points show where the dataset is densely populated.

Direct Density Reachability: If a point q is a core point and a data point p is in its ϵ *-neighborhoo*d, then p is directly density-reachable from q.

Density-Connected Components: A maximum collection of data points in which each point is directly density-reachable from each other point in the same set is known as a density-connected component.

Noise Points: Outliers or noise points are data points that do not fit into any density-connected component.

DBSCAN uses the aforementioned concepts to

categorize data points as noise, outliers, or clusters. The approach adds core points and their reachable neighbors iteratively, starting with an arbitrary data point and growing the cluster. Until no more core points can be added to the cluster, the procedure is repeated.

3. Deep learning algorithms

3.1 DNN algorithm

The field of artificial intelligence and machine learning has seen a radical transformation in recent years with the introduction of Deep Neural Networks. DNNs are a type of artificial neural network that can extract hierarchical representations from input. They are made up of several layers. Deep neural networks (DNNs) are far more complicated task-aware than standard shallow networks because they automatically find complex patterns in the data. Examples of these tasks include picture and audio recognition, natural language processing, and reinforcement learning.

The fundamental component of DNNs is its design, which is made up of three layers: input, hidden, and output. Each layer is made up of linked nodes or neurons. Through a process known as backpropagation, the weights associated with the connections between neurons are learnt, and these weights are then adjusted to minimize the discrepancy between the network's predictions and the actual targets. The following are the main theoretical ideas that underpin DNNs:

Activation Functions: To incorporate non-linearities and enable the network to learn and represent complicated connections within the data, neurons inside DNNs use activation functions, such as the sigmoid or rectified linear unit (ReLU).

Feedforward and Backpropagation: DNNs handle data in a feedforward fashion. During training, errors are computed using a loss function. Iteratively enhancing the network's performance, the back-propagation technique modifies the network's weights to minimize this error.

Optimization approaches: To improve training and avoid overfitting, DNNs use a variety of optimization approaches, including as stochastic gradient descent, adaptive learning rates (like Adam, RMSprop), and regularization techniques (like dropout).

DNNs are fundamentally based on their capacity to extract complex characteristics from data. Every layer in the network picks up increasingly complicated representations as data moves across it. DNNs are able to abstract features through hierarchical representation learning, which gives them the capacity to identify patterns in unpro-cessed data. This skill is critical for jobs like object identification, language translation, and decision-making.

3.2 ResNet algorithm

Although deep neural networks have shown impressive results in a variety of applications, they encounter major challenges as network depth grows. The vanishing gradient problem makes deep networks harder to train and makes learning meaningful representations in very deep architectures more complex. He et al. (2015) presented Residual Networks (ResNets)[10] as a clever solution to this issue, making it possible to build astonishingly deep networks that are more accurate and easier to train. The insertion of residual connections, which provide shortcut pathways for the gradient to travel more directly from the input to the output, is the core innovation of ResNets. In contrast, every layer in classic feedforward designs has to learn an identity mapping. The following are the main theoretical ideas that underpin ResNets:

Residual Learning: ResNets present the idea of residual blocks, which comprise one or more convolutional layers in each block. Relative blocks try to learn the residual, or the difference between the intended output and the current input, as opposed to learning the output directly. The final output is then obtained by adding the identity mapping to the residual.

Shortcut Connections: provide gradients a straight route to follow without passing through the intermediary layers. Gradients can now avoid the vanishing gradient issue, allowing for the efficient training of extremely deep networks.

Stochastic Gradient Descent with ResNets: By having shortcut connections, gradients may propagate along a clear path during backpropagation, which helps to mitigate the vanishing gradient issue that usually causes training deep networks to be difficult.

3.3 LSTM algorithm

Sequential data, which is distinguished by its temporal dependencies, is widely used in real-world applications, such as financial time series, sensor data, audio, and text. Recurrent neural networks (RNNs) of the Long Short-Term Memory (LSTM) [11] kind have addressed the drawbacks of conventional RNNs and become an essential tool for modeling sequential data.

The primary advancement in LSTMs is the addition of memory cells, which allow them to retain information across lengthy periods and detect long-range relationships. Among the fundamental ideas in LSTM theory are:

Memory Cells: LSTMs are equipped with memory cells that have the capacity to store and retrieve data across several time steps. Three gates are included in these cells: an input gate, an output gate, and a forget gate. The information entering and leaving the memory cell is managed by these gates.

Input Gate: This device controls how information is updated into the memory cell. It chooses which data from the previous cell state and the current input is kept in the cell.

Forget Gate: The forget gate regulates whether data is kept in the cell's memory or deleted. It makes the decision about what data from the prior cell state should be retained and what should be deleted.

Output Gate: At each time step, the output gate determines whether data from the memory cell should be sent to the network's output.

Activation Functions: To regulate the information flow within the memory cell and gates, latch-state transistors (LSTMs) employ activation functions such as the hyperbolic tangent (tanh).

Ⅲ. Experiment setup and Result

1. Experiment

Retina-4SN radar is an FMCW radar that we employ in this work to collect experiment data, as shown in figure 2. It can obtain point cloud data in four dimensions (x, y, z, and Doppler velocity) via a WiFi interface. It can transmit an FMCW signal at frequencies ranging from 71 to 81 GHz.

Fig. 2. Retina-4SN radar.

Three different-sized drones are used to collect drone data. Additionally, we enlisted three people, each with a different height and body composition, to collect human data. The experiment is conducted indoors in a multi sport stadium. The dataset was acquired inside a 7m (Ox) x 5.5m (Oy) rectangle.

Three primary scenarios comprise the data collecting process: the first involves a monitoring region with a single item which is a human, the second involves a tracking area with single item which a drone, and the third involves more than one thing. A person or drone will be transported along various fundamental trajectories such as: freely, in a circle, vertically, horizontally, and diagonally. When there are two or more objects,

they are moved along the same trajectory as if there were only one. We are also interested in the objects' relative positions to one another, such as when the first object is on the left, right, front, or behind the second object, as well as in unique situations like when the drone flies directly overhead a person.

2. Result and Discussion

2.1 Clustering algorithm

In this paper, we choose the ϵ value based on the analyzes distances between pairs of points in three-dimensional (3D) coordinate spaces. Initially, we compute the distances between pairs of points in the 3D (Oxyz) coordinate system. Following that, we detect peak (ϵ_{3D}) in distance distributions of three dimension. Within each static frame, the DBSCAN algorithm employs the 3D coordinates of all data points (C_{3D}) as well as ϵ_{3D} value to determine the cluster count and to assign each point to its respective cluster. The accuracy (A_c) is calculated using the below equation:

$$A_c = \frac{TF}{TF + FF} \quad (1)$$

where:

TF: the number of frames which are correctly clustered

FF: the number of frames which are incorrectly clustered

Table 1 shows the result when using DBSCAN algorithm in clustering frame in three cases such as: only 1 drone, only 1 human and more than one object in tracking area. When the data table is examined, it becomes evident that the scenario involving only a single drone attains the highest accuracy, reaching 92.8% across 21,368 frames. The second-best accuracy of 88.87% is observed in the case of a single drone over 13,193 frames. In scenarios where there are 2 or more objects in the frames, the lowest clustering accuracy is achieved, amounting to 71.87% across 39,987 frames. During our observations, the drone's point clouds exhibit a uniform and concentrated density, forming cohesive blocks. In contrast, the point clouds generated by individuals occasionally lack such cohesion, often splitting into two or three distinct segments, based on the person's height. This fragmentation presents a challenge for DBSCAN, causing confusion during the clustering process. Furthermore, when multiple objects are in close proximity, with a distance less than 37cm (± 3cm) between them, their respective point clouds also converge, leading to DBSCAN mistakenly grouping them into the same cluster.

Table 1. Accuracy of clustering result by using DBSCAN algorithm.

	Drone	Human	Multiple
Accuracy	92.8%	88.87%	71.87%
Total	21368	13193	39987

2.2 Standardizing data

We normalize the original data for each cluster in the left column of table 2 into the appropriate input characteristics in the right column before training the model.

Table 2. Original data and Input features of a cluster.

Original data	Input features
3D Coordinates of all points (C)	Number of points (N_p)
	centroid point x (c_x)
	centroid point y (c_y)
	centroid point z (c_z)
	D_x
	D_y
	D_z
Powers of all points	Average power (avg_p)
Velocities of all points	Average velocity (avg_v)
Label	
Human	1
Drone	0
Noise	-1

First, the 3D coordinates of n points (C): ($(x_1, y_1, z_1), (x_2,y_2,z_2), \ldots, (x_n,y_n,z_n)$) will be transformed into N_p = n, c_x calculated using equation (2), c_y calculated using equation (3), and c_z calculated using equation (4). Additionally, D_x, D_x, and D_x will be determined using equations (5), (6), and (7), respectively.

$$c_x = \frac{\sum_{k=1}^{n} x_k}{n} \tag{2}$$

$$c_y = \frac{\sum_{k=1}^{n} y_k}{n} \tag{3}$$

$$c_z = \frac{\sum_{k=1}^{n} z_k}{n} \tag{4}$$

$$D_x = \max(x_1,\ldots,x_n) - \min(x_1,\ldots,x_n) \tag{5}$$

$$D_y = \max(y_1,\ldots,y_n) - \min(y_1,\ldots,y_n) \tag{6}$$

$$D_z = \max(z_1,\ldots,z_n) - \min(z_1,\ldots,z_n) \tag{7}$$

In the process of cluster labeling, we allocated numerical values to each cluster: we used -1 to represent noise, 0 to indicate drones, and 1 to signify humans. The power values of n points $(p_1, p_2, \ldots, p_n)$ are transformed into the avg_p, and computed using equation (8):

$$avg_p = \frac{\sum_{k=1}^{n} p_k}{n} \tag{8}$$

Finally, the velocity of n points $(v_1, v_2, \ldots, v_n)$ are normalized into the avg_v, and calculated using equation below:

$$avg_v = \frac{\sum_{k=1}^{n} v_k}{n} \tag{9}$$

2.3 Object detection

The input data for three deep learning models comprises nine input features (N_p, c_x, c_y, c_z, D_x, D_y, D_z, avg_p, avg_v) which are described in table 2. The architecture of our DNN model is shown in Figure 3(a). It features a Batch Normalization layer at the start to standardize the input data, along with a dropout layer to mitigate overfitting. Figure 3(b) illustrates the architecture of the ResNet model used in our research, which closely resembles the DNN model. This configuration includes two hidden layers, each preceded by a Batch Normalization layer to normalize the input data and apply the rectified linear unit (ReLU) activation layer. In the second block, we've incorporated a shortcut link that adds the intermediate input to its output before passing through the ReLU layer. This approach helps address the common issue of vanishing gradients often encountered in DNN algorithms. Due to the small number of nine input features, in the ResNet model training phase, the effort to boost accuracy by adding more residual blocks did not yield improvements. Additionally, this adjustment significantly extended the training duration. As a result, a choice was made to adopt a simplified approach, employing just one hidden layer and a single residual block. To combat overfitting, a dropout layer is applied

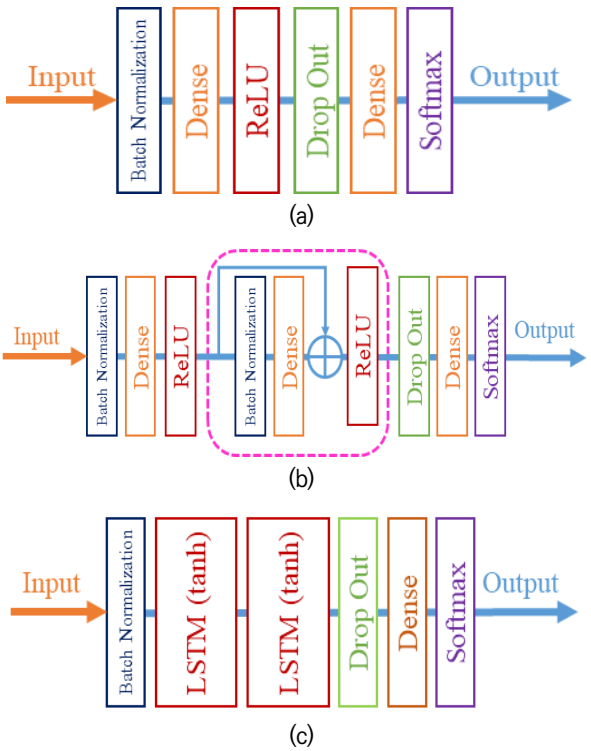


Fig. 3. (a) DNN architecture, (b) ResNet architecture, (c) LSTM architecture.

after the second layer.

Moving on to our LSTM network model, as depicted in Figure 3(c), it comprises two LSTM layers. We also implement Batch Normalization to preprocess the input data before it enters these layers, and we use dropout to counteract overfitting.

The dataset comprises input features from 63,252 clusters representing humans and 82,940 clusters representing drones. These data are then used to evaluate the accuracy of three different deep learning models in terms of their outputs. The precise ratio (A_d) is computed using the formula provided below.

$$A_d = \frac{TC}{TC + FC} \tag{10}$$

where:

TC: the number of clusters which are correctly predicted

FC: the number of clusters which are wrongly predicted

The detailed accuracy data of the three models in cluster prediction is illustrated in the figure 4. It's evident that the LSTM model demonstrates the lowest accuracy among the three models, specifically, 78.94% for humans and 75.63% for drones.

On the other hand, the ResNet and DNN models exhibit commendable performance in predicting both people and drones. However, across the board, ResNet consistently delivers superior results. It achieves 98.62% accuracy for drones and 96.75% for humans. Comparatively, DNN presents an accuracy of 98.49% for drones, a marginal 0.13% lower than ResNet. Nevertheless, the predictions for people with the DNN model yield lower results, standing at 94.07%, which is 2.68% less than ResNet's accuracy in this category.

Furthermore, the Retina-4SN radar has its own application designed for tracking people and recognizing human movements. To assess its performance, we conducted a comparative analysis

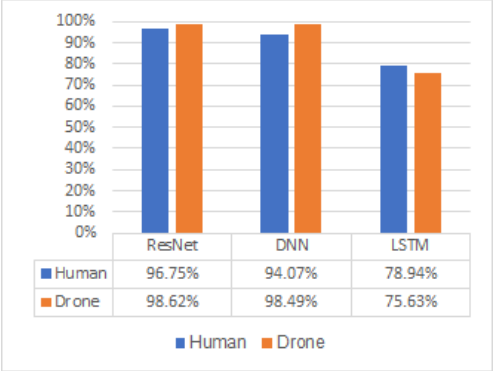
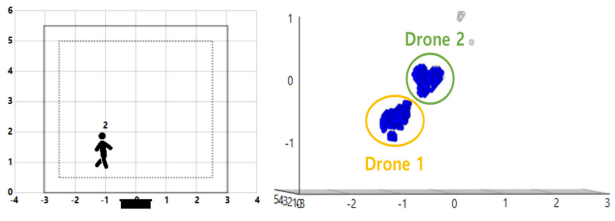
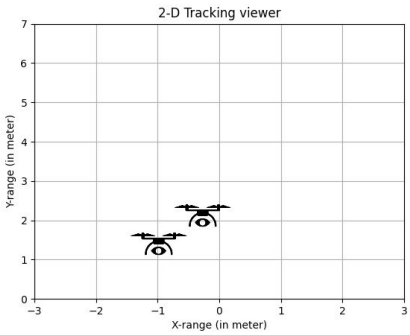


Fig. 4. Comparison accuracy of ResNet, DNN, LSTM models in detection object.

with our ResNet model. In certain scenarios, the integration of the ResNet model yielded more precise results. For instance, as depicted in Figure 5(a), when two drones closely fly together, the radar system occasionally misidentifies them as a specific person, as observed in the tracking viewer on the left. In contrast, our system's tracking viewer, shown in Figure 5(b), provides accurate results in the case of two drones flying in close proximity.



(a)



(b)

Fig. 5. (a) tracking view result and raw point cloud data in a static frame from Retina-4SN's application, (b) our tracking view result in same frame with (a).

Another scenario in Figure 6(a) involves a situation where drone 1 is flying near the ground.

In this instance, the radar system mistakenly categorizes it as a person, while our system's tracking viewer correctly identifies it as two drones as shown in Figure 6(b).

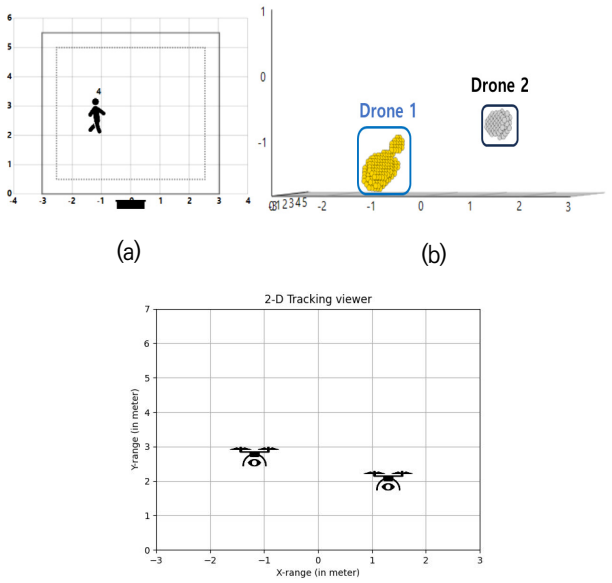


Fig. 6. (a) tracking view result and raw point cloud data from Retina-4SN's application when a drone flies near the ground, (b) our tracking view result in same frame with (a).

2.4 Object tracking

In order to establish the object's location within the tracking area, we make use of two input features, specifically c_x and c_y, as indicated in the table 2. These parameters represent the object's coordinates along the Ox and Oy axes. Subsequently, following the identification of the object as either a human or a drone through the deep learning model, it is depicted on the 2D tracking viewer, with the corresponding icon positioned at the coordinates (c_x, c_y).

In Figure 7, we observe a scenario where the tracking area contains multiple distinct objects, comprising three individuals and two drones. Figures 7(a) and 7(b) present the raw data captured by the radar in a three-dimensional space, offering views from the front and from above, respectively. Figure 7(c) represents the tracking viewer generated by our system. Upon closer examination, it becomes evident that Figure 7(c) provides a comprehensive display of all object types and their precise positions, in direct comparison to the experimental setup.

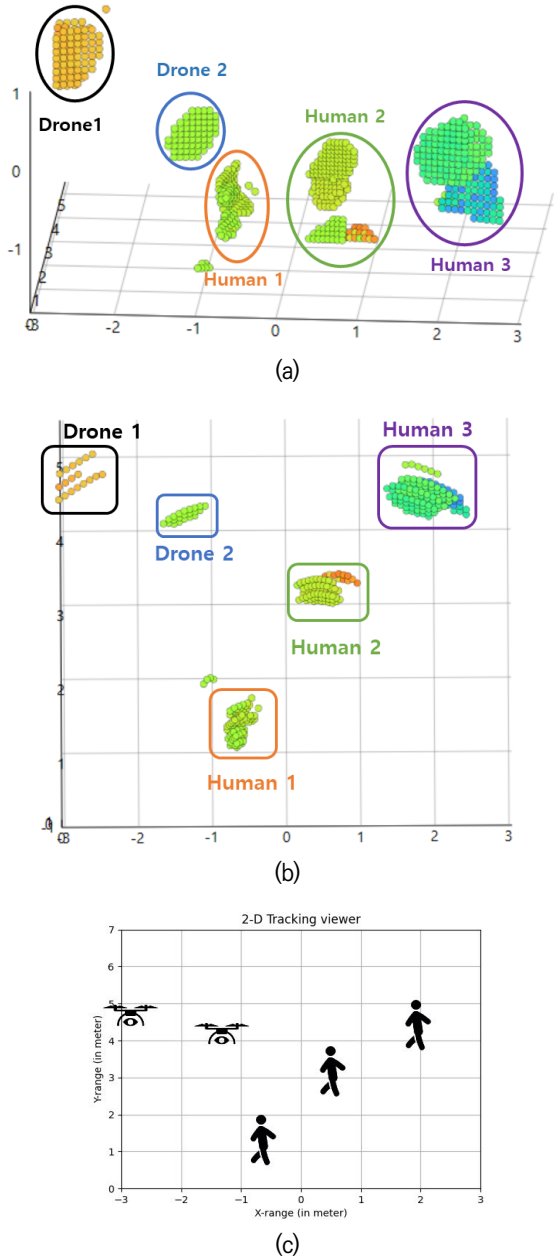


Fig. 7. (a), (b) tracking view result and raw point cloud data from Retina-4SN's application, (c) our tracking view result in same frame with (a), (b).

Ⅳ. Conclusion and Future work

1. Conclusion

This research paper introduces a system designed to detect and track humans and drones based on

data acquired from FMCW radar. Our approach relies on the data from point clouds, which includes coordinate information, along with associated power and velocity values, as input data. Subsequently, we employ DBSCAN for clustering and segmenting the points into respective clusters within the point cloud. The data is then normalized, and a deep learning model is utilized to predict object clusters.

An examination of the clustering results reveals that DBSCAN performs most effectively when the tracking area contains only one object, particularly when that object is a specific drone, yielding an accuracy of up to 92.8%. However, when applied to frames with two or more objects, the accuracy of this algorithm substantially diminishes, highlighting its limitations.

Regarding object detection, after employing three models with the dataset, the findings indicate that both DNN and ResNet models provide outputs with commendable accuracy. In particular, when analyzing specific data, the ResNet model excels in predicting clusters of drones and specific individuals. It achieves an accuracy rate of 96.75% for people and 98.62% for drones.

2. Future work

Upon reviewing the data table reflecting the results of DBSCAN's clustering, it becomes evident that this algorithm has notable limitations, particularly in tracking areas featuring multiple distinct objects. Only 71.87% of static frames are correctly clustered. Two primary reasons contribute to this challenge: firstly, when two clusters are positioned in close proximity, within a distance shorter than 37cm (±2cm), DBSCAN tends to misidentify them as a single entity. The second reason involves human clusters, which sporadically appear. Instances where points exhibit uneven density distribution, forming two or three separate small groups, also lead to incorrect outcomes when employing DBSCAN.

To address these limitations, our plans involve researching mathematical theories to enhance the DBSCAN algorithm, tailoring it to better suit our dataset, to mitigate the aforementioned errors and enhance clustering accuracy. Simultaneously, we aim to continually gather additional experimental data to further refine our deep learning model.

References

[1] Ahirwar, S and Swarnkar, R and Bhukya, S and Namwade, G, "Application of drone in agriculture," *International Journal of Current Microbiology and Applied Sciences,* vol.8, no.1, pp.2500-2505, 2019.
DOI: 10.20546/ijcmas.2019.801.264

[2] Son, Seung Woo and Yu, Jae Jin and Kim, Dong Woo and Park, Hyun Su and Yoon, Jeong Ho, "Applications of drones for environmental monitoring of pollutant-emitting facilities," *Proceedings of NIE,* vol.2, no.4, pp.298-304, 2021.
DOI: 10.22920/PNIE.2021.2.4.298

[3] Fan, Jin and Saadeghvaziri, M Ala, "Applications of drones in infrastructures: Challenges and opportunities," *International Journal of Mechanical and Mechatronics Engineering,* vol.13, no.10, pp.649-655, 2019. DOI: 10.5281/zenodo.3566281

[4] Björklund, S, "Target detection and classification of small drones by boosting on radar micro-Doppler," in *Proc. of 2018 15th European Radar Conference (EuRAD),* pp.182-185, 2018.
DOI: 10.23919/EuRAD.2018.8546569

[5] Jian, Michael and Lu, Zhenzhong and Chen, Victor C, "Drone detection and tracking based on phase-interferometric Doppler radar," in *Proc. of 2018 IEEE Radar Conference (RadarConf18),* pp.1146-1149, 2018.
DOI: 10.1109/RADAR.2018.8378723

[6] Park, J., Jung, D. H., Bae, K. B., & Park, S. O, "Range-Doppler map improvement in FMCW radar for small moving drone detection using the stationary point concentration technique," *IEEE Transactions on Microwave Theory and Techniques,*

vol.68, no.5, pp.1858-1871, 2020.
DOI: 10.1109/TMTT.2019.2961911
[7] Companik, E., Gravier, M. J., & Farris II, M. T, "Feasibility of warehouse drone adoption and implementation," *Journal of Transportation Management,* vol.28, no.2, pp.5, 2018.
DOI: 10.22237/jotm/1541030640
[8] Wawrla, L., Maghazei, O., & Netland, T, "Applications of drones in warehouse operations," *Whitepaper. ETH Zurich, D-MTEC*, pp.212, 2019.
[9] Ester, Martin, et al, "A density-based algorithm for discovering clusters in large spatial databases with noise," *kdd,* vol.96, no. 34, pp.226-231, 1996.
[10] He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian, "Deep residual learning for image recognition," in *Proc. of the IEEE conference on computer vision and pattern recognition,* pp.770-778, 2016.
DOI: 10.1109/CVPR.2016.90
[11] Hochreiter, Sepp, and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol.9, no.8, pp.1735-1780, 1997.
DOI: 10.1162/neco.1997.9.8.1735

BIOGRAPHY

Ha Thi Tran (Member)

2019 : BE degree in Information and Technology, University of Engineering and Technology, Vietnam National University
2023~current : M.S. degree in Information and Communication Engineering, Kongju National University.

The-Hien Pham (Member)

2012 : BE degree in Communication and Networking, Ho Chi Minh city University of Transport, Vietnam.
2022 : M.S. degree in Information and Communication Engineering, Kongju National University.
2023~current : Ph.D. degree in Information and Communication Engineering, Kongju National University.

Yun-Seok Mun (Member)

2018~Current : BS degree in Smart Information Technology Engineering, Kongju National University.

Ic-Pyo Hong (Member)

1994 : BS degree in Electronics Engineering, Yonsei University.
1996 : MS degree in Electronics Engineering, Yonsei University.
2000 : PhD degree in Electronics Engineering, Yonsei University.
2000~2003 : Senior Engineer in CDMA Mobile Research, Samsung Electronics.
2006 : Visiting Scholar with Texas A&M University.
2012 : Visiting Scholar with Syracuse University.
2003~current : Professor, Department of Information and Communication Engineering, Kongju National University.