

Article

Simultaneous Tracking and Recognizing Drone Targets with Millimeter-Wave Radar and Convolutional Neural Network

Suhare Solaiman * , Emad Alsuwat  and Rajwa Alharthi

Department of Computer Sciences, College of Computers and Information Technology, Taif University,
Taif 26571, Saudi Arabia

* Correspondence: s.m.solaiman@tu.edu.sa

Abstract: In this paper, a framework for simultaneous tracking and recognizing drone targets using a low-cost and small-sized millimeter-wave radar is presented. The radar collects the reflected signals of multiple targets in the field of view, including drone and non-drone targets. The analysis of the received signals allows multiple targets to be distinguished because of their different reflection patterns. The proposed framework consists of four processes: signal processing, cloud point clustering, target tracking, and target recognition. Signal processing translates the raw collected signals into sparse cloud points. These points are merged into several clusters, each representing a single target in three-dimensional space. Target tracking estimates the new location of each detected target. A novel convolutional neural network model was designed to extract and recognize the features of drone and non-drone targets. For the performance evaluation, a dataset collected with an IWR6843ISK mmWave sensor by Texas Instruments was used for training and testing the convolutional neural network. The proposed recognition model achieved accuracies of 98.4% and 98.1% for one and two targets, respectively.

Keywords: mmWave radar; cloud points; target tracking; target recognition



Citation: Solaiman, S.; Alsuwat, E.; Alharthi, R. Simultaneous Tracking and Recognizing Drone Targets with Millimeter-Wave Radar and Convolutional Neural Network. *Appl. Syst. Innov.* **2023**, *6*, 68. <https://doi.org/10.3390/asi6040068>

Received: 30 May 2023

Revised: 19 July 2023

Accepted: 31 July 2023

Published: 1 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, unmanned aerial vehicles (UAVs), such as drones, have received significant attention for performing tasks in different domains. This is because of their low cost, high coverage, and vast mobility, as well as their capability to perform different operations using small-scale sensors [1]. Smartphones can now operate drones instead of traditional remote controllers, owing to technological advancements. In addition, drone technology can provide live video streaming and image capturing, as well as make autonomous decisions based on these data. Consequently, artificial intelligence techniques have been utilized in the provisioning of civilian and military services [2]. In this context, drones have been adopted for express shipping and delivery [3–5], natural disaster prevention [6,7], geographical mapping [8], search and rescue operations [9], aerial photography for journalism and film making [10], providing essential materials [11], border control surveillance [12], and building safety inspection [13]. Even though drone technology offers a multitude of benefits, it raises mixed concerns when it comes to how it will be used in the future. Drones pose many potential threats, including invasion of privacy, smuggling, espionage, flight disruption, human injury, and terrorist attacks. These threats compromise aviation operations and public safety. However, it has become increasingly necessary to detect, track, and recognize drone targets and make decisions in certain situations, such as detonating or jamming unwanted drone targets.

The detection of unwanted drones poses significant challenges to observation systems, especially in urban areas, as drones are tiny and move at different rates and heights compared to other moving targets [2]. For target recognition, optic-based systems that rely on cameras provide more detailed information than radio-frequency (RF)-based systems,

but these require a clear frontal view, as well as ideal light and weather conditions [14,15], as shown in Figure 1A. Both residential and business environments are less accepting of the use of cameras for target recognition because of their intrusive nature [16]. Although RF-based systems are less intrusive, the signals received from RF devices are not as expressive or intuitive as those received from images. Humans are often unable to directly interpret RF signals. Thus, preprocessing RF signals is a challenging process that requires the translation of raw data into intuitive information for target recognition. It has been proven that RF-based systems such as WiFi, ultrasound sensors, and millimeter-wave (mmWave) radar can be useful for a variety of observation applications that are not affected by light or weather conditions [17]. WiFi signals require a delicate transmitter and receiver and are limited to situations where targets must move between the transmitter and receiver [18]. Because ultrasound signals are short-range, they are usually used to detect close targets and are affected by blocking or interference from other nearby transmitters [14].

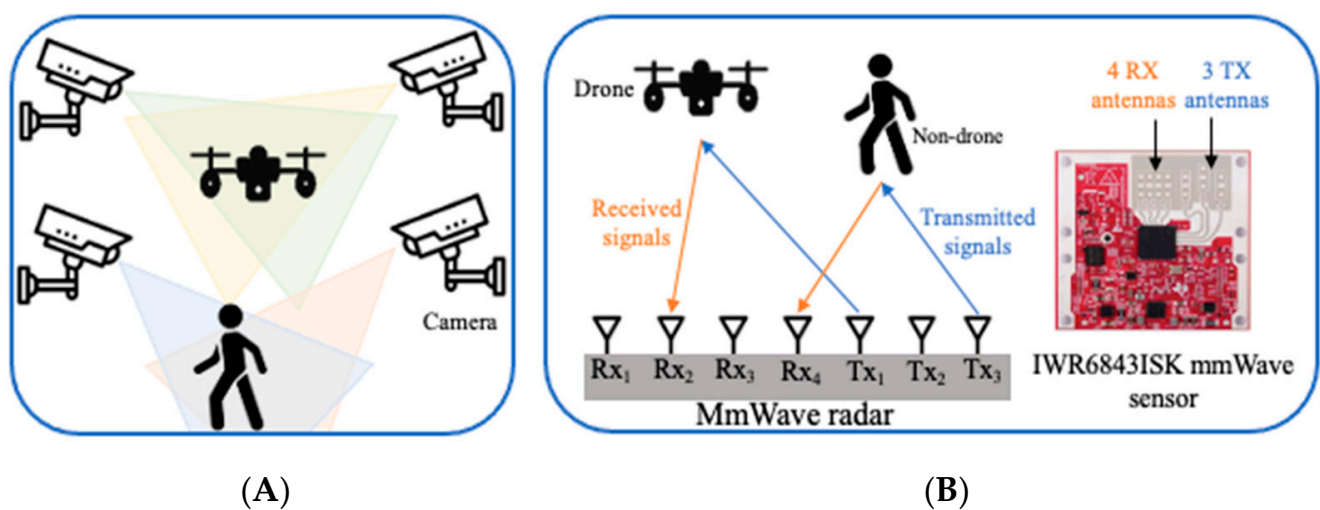


Figure 1. In contrast to (A), an optic-based system, (B) the proposed framework is based on mmWave radar, which consists of three transmitting antennas and four receiving antennas.

The large bandwidth of mmWave allows a high distance-independent resolution, which not only facilitates the detection and tracking of moving targets, but also their recognition [18]. Furthermore, mmWave radar requires at least two antennas for transmitting and receiving signals; thus, the collected signals can be used in multiple observation operations [18]. Rather than true color image representation, mmWave signals can represent multiple targets using reflected three-dimensional (3D) cloud points, micro-Doppler signatures, RF-intensity-based spatial heat maps, or range-Doppler localizations [19].

mmWave-based systems frequently use convolutional neural networks (CNNs) to extract representative features from micro-Doppler signatures to recognize objects [18,20,21]; however, examining micro-Doppler signals is computationally complex because they deal with images, and they only distinguish moving targets based on translational motion. Employing a CNN to extract representative features from cloud points is becoming the tool of choice for developing the mathematical modeling underlying dynamic environments and leveraging spatiotemporal information processed from range, velocity, and angle information, thereby improving robustness, reliability, and detection accuracy and reducing computing complexity to achieve the simultaneous performance of mmWave radar operations [22].

To solve these challenges, a novel framework for the simultaneous tracking and recognition of drone targets using mmWave radar is proposed. The proposed framework is based on the installation of a low-cost and small-sized mmWave sensor for transmitting and receiving signals, as shown in Figure 1B. Our main objective was to utilize 3D point clouds generated by a mmWave radar to detect, track, and recognize multiple moving

targets, including drone and non-drone targets. When raw analog-to-digital conversion data from antenna arrays are converted into 3D point clouds, the data size is reduced from tens of gigabytes to a few megabytes [23]. This allows for the faster data transfer, processing, and application of complex machine learning algorithms. Unlike the micro-Doppler signature, the spatiotemporal features of cloud points are more representative and easily interpretable because movements occur in a 3D space. For performance evaluation, a dataset (<https://github.com/00Nave198/MSU-ECE480-13-mmWave/tree/main/data>, accessed on 1 August 2023) collected with an IWR6843ISK mmWave sensor by Texas Instruments (TI) was used for training and testing the CNN. Our key contributions can be summarized as follows:

1. A signal-processing algorithm is proposed to generate 3D point clouds of multiple moving targets in the field of view (FoV), considering both static and dynamic reflections of mmWave radar signals.
2. A multitarget tracking algorithm was developed that integrates a density-based clustering algorithm to merge related point clouds into clusters, extended Kalman filters (EKF) to estimate the new position of the detected targets, and the Hungarian algorithm to match each new estimated track with its target cluster.
3. A novel CNN model is proposed for feature extraction and identification of drone and non-drone targets from clustered 3D cloud points.
4. The performance of the proposed tracking and recognition algorithms was evaluated.

The rest of this paper is organized as follows. Section 2 reviews the existing literature. In Section 3, the proposed framework is presented. Section 4 describes the signal preprocessing. The clustering process is described in Section 5. The tracking process is described in Section 6. Section 7 describes the recognition process. Section 8 discusses the performance evaluation. Section 9 presents conclusions and suggestions for future work.

2. Related Works

Several techniques have been developed to detect and recognize drones, including visual [24], audio [25], WiFi [26,27], infrared camera [28], and radar [29]. Drone audio detection relies on detecting propeller sounds and separating them from the background noise. A high-resolution daylight camera and a low-resolution infrared camera were used for visual assessment [30]. Good weather conditions and a reasonable distance between drone targets and cameras are still required for visual assessment. Fixed visual detection methods cannot estimate the continuance track of drones. Infrared cameras detect heat sources on drones such as batteries, motors, and motor driver boards. Airborne vehicles can be detected more easily by mmWave radar, which has been the most-popular form of detection for military troops for a long time. However, traditional military radars are designed to recognize large targets and have trouble detecting small drones. Furthermore, the target discrimination may not be straightforward. The extremely short wavelength of mmWave radar systems makes them highly sensitive to the small features of drones, providing very precise velocity resolution, and allowing them to penetrate certain materials to detect concealed hazardous targets [30].

This subsection discusses various recent drone classifications using machine learning and deep learning models. The radar cross-section (RCS) signatures of different drones with different frequency levels have been discussed in several studies, including [2,31]. The method proposed in [2] relied on converting the RCS into images and then using a CNN to perform drone classification, which required much computation. As a result, they introduced a weight-optimization model that reduces the computational overhead, resulting in improved long short-term memory (LSTM) networks. The authors showed how a database of mmWave radar RCS signatures can be utilized to recognize and categorize drones in [31]. They demonstrated RCS measurements at 28 GHz for a carbon-fiber drone model. The measurements were collected in an anechoic chamber and provided significant information regarding the RCS signature of the drone. The authors aided the RCS-based detection probability and range accuracy by performing simulations in metropolitan environments.

The drones were placed at different distances ranging from 30 m to 90 m, and the RCS signatures used for detection and classification were developed by trial and error.

The authors proposed a novel drone-localization and activity-classification method using vertically oriented mmWave radar antennas to measure the elevation angle of the drone from the ground station in [32]. The measured radial distance and elevation angle were used to estimate the height of the drone and the horizontal distance from the radar station. A machine learning model was used to classify the drone's activity based on micro-Doppler signatures extracted from radar measurements taken in an outdoor environment.

The system architecture and performance of the FAROS-E 77 GHz radar at the University of St Andrews were reported in [33] for detecting and classifying drones. The goal of the system was to demonstrate that a highly reliable drone-classification sensor could be used for security surveillance in a small, low-cost, and portable package. To enable robust micro-Doppler signature analysis and classification, the low phase noise and coherent architecture take advantage of the high Doppler sensitivity available at mmWave frequencies. Even when a drone hovered in a stationary manner, the classification algorithm was able to classify its presence. In [34], the authors employed a vector network analyzer that functioned as a continuous wave radar with a carrier frequency of 6 GHz to gather Doppler patterns from test data and then recognize the motions using a CNN.

Furthermore, the authors of [35] proposed a method for the registration of light detection and ranging (LiDAR) point clouds and images collected by low-cost drones to integrate spectral and geometrical data.

3. The Proposed Framework

The proposed framework for the simultaneous tracking and recognition of drone targets using mmWave radar is presented in this section. A front-end mmWave radar system with three transmitting antennas (Tx) and four receiving antennas (Rx) is shown in Figure 1. The mmWave radar transmits multiple frequency-modulated continuous waveform (FMCW) chirps. These signals are received by the receiving antennas after being reflected from multiple targets in the FoV. The radar then combines the Tx and Rx signals to demodulate the FMCW signals and generate intermediate-frequency (IF) signals, creating a time-stamped snapshot of the FoV [36]. The collected sequence of IF signals is insufficiently informative and, hence, was applied to preliminary preprocessing to extract some features of the target, such as the range, velocity, and angle [20,36]. Figure 2 shows the proposed framework, which consists of four modules, which operate using a pipelined approach as follows:

5. Signal preprocessing: This module translates the raw information collected by the mmWave radar into sparse point clouds and eliminates the points associated with interference noises and static objects (i.e., points that appeared in the previous frame), which reveal the existence and movement of targets.
6. Clustering: This module detects different moving targets and merges the related point clouds into clusters, where each cluster represents a single moving target.
7. Tracking: This module estimates a target track in successive frames and applies an association algorithm to track multiple targets' paths.
8. Recognition: This module utilizes a CNN model to extract representative spatiotemporal features from cloud points and then classifies the detected targets as drone and non-drone targets.

The four modules of the proposed framework are explored in detail in the following sections.

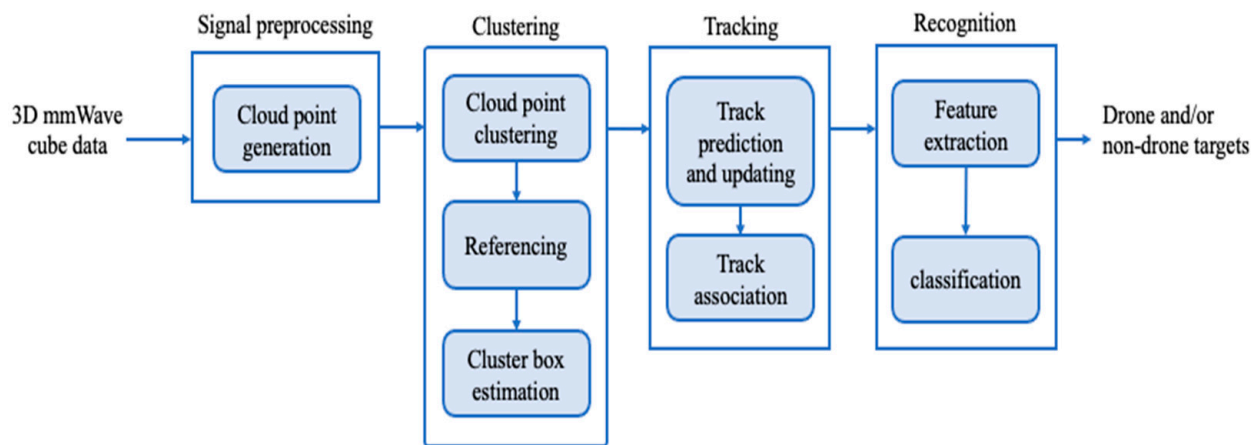


Figure 2. The proposed framework.

4. Signal Preprocessing

The raw IF signals were collected in the form of a 3D cube data (time, chirp, and antenna). Fast Fourier transformations were performed on the IF signals to estimate the range, velocity, and angle of arrival (AoA) of the moving target [14,37]. A cloud point is a 3D model composed of a set of points used in the literature to describe a list of detected targets provided by radar signal processing [38]. Figure 3 illustrates a four-step signal preprocessing workflow to generate a series of cloud points, each comprising different features, such as the 3D spatial position (x, y, and z), velocity, and AoA [39].

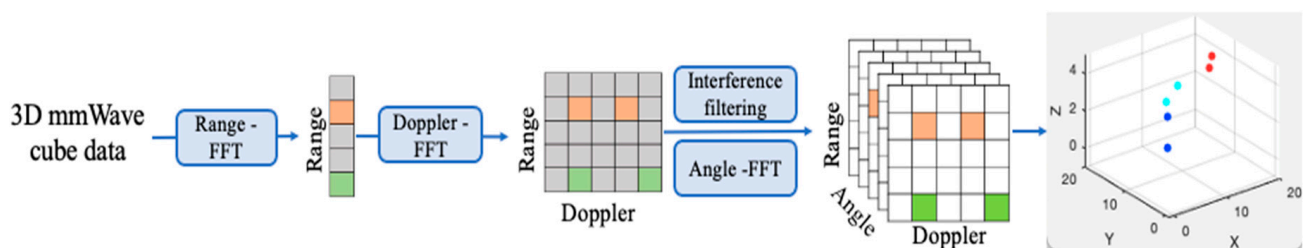


Figure 3. The workflow of signal processing and cloud point generation.

4.1. Range Fast Fourier Transform

FMCW-transmitted chirps are characterized by the frequency f , bandwidth b , and duration T_{chirp} . The reflected IF signal is parsed to determine the radial range between the radar and the target. The frequency of IF signal f_{IF} is proportional to the radial distance r and is denoted as:

$$f_{IF} = \frac{s2r}{c}, \quad (1)$$

However, the radial distance between the radar and target is estimated as:

$$r = \frac{f_{IF}c}{2s}, \quad (2)$$

where c represents the speed of light 3×10^8 m/s and s is the chirp frequency slope, which is calculated as $(s = \frac{b}{T_{chirp}})$. The range-fast Fourier transform (FFT) is applied to each chirp of the radar data cube to convert the time domain IF signal into the frequency domain. The peak of the resulting frequency spectrum determines the range of each target. The distance can be calculated by averaging the distance collected by all the chips in a frame and the number of chirps in the frame.

4.2. Doppler Fast Fourier Transform

A slight change in the distance to the target resulted in a significant shift in the IF signal phase. To determine the target velocity, chirps separated by two or more times in duration T_{chirp} are required. Subsequently, a Doppler-FFT was applied across the phases received from these chirps. Therefore, the target radial velocity can be estimated by comparing the phase differences between two received signals. If the target is moving, the phase difference ω can be calculated as:

$$\omega = \frac{4\pi T_{chirp} v}{\lambda}, \quad (3)$$

This approach can discriminate between targets moving at various velocities and at the same distance. The target velocity v for each moving target can be calculated as:

$$v = \frac{\lambda \omega}{4\pi T_{chirp}}, \quad (4)$$

where λ is the wavelength.

The phase difference between the two chirps at the range-FFT peak is proportional to the radial velocity of the detected target. Applying the Doppler-FFT to the signal range spectrum yielded 2D range-Doppler localizations.

4.3. Interference Filtering

Interference filtering is responsible for removing the interference scattering points reflected from unwanted objects in the FoV. Reflections from a noisy background, such as reflections from walls, must be removed, as well as reflections from the clutter of static (non-moving) objects, such as trees. Because drone targets are continuously moving, reflections from drone targets are combined with such inferences, causing significant issues in the clustering, tracking, and recognition processes of drones. The interference is filtered by applying a constant false alarm rate (CFAR) [40] and the moving target indication (MTI) [41].

4.3.1. CFAR Algorithm

The received signal X_r^t from the FoV can be expressed as:

$$X_r^t = X_s^t + g^t, \quad (5)$$

where X_s^t is the target reflection and g^t is the white Gaussian noise in a certain frame t .

From the received signal, the CFAR algorithm [42] is applied to detect the presence or absence of the target. A fixed threshold value is used in traditional detectors such as the Neyman–Pearson detector [38]. The assumption is that interference (noise or clutter) is spread similarly across the test range bins such that, if the signal in the test bin exceeds a specific threshold γ , the bin contains a target. This results in false alarm conditions, as shown in the following equations:

$$X_s^t + g^t \geq \gamma, \text{ true detection} \quad (6)$$

$$g^t \geq \gamma, \text{ false alarm} \quad (7)$$

A CFAR detector maintains a constant false alarm rate, which adjusts the detection threshold within the range bins. The detector calculates the noise level inside a sliding window and uses this estimate to assess the presence or absence of a target in the test bin. If a target is found in a bin, the algorithm returns the target range-Doppler localization [38]. Finally, all CFAR-identified targets are organized into groups based on their positions in a 3D matrix. In certain cases, this assumption might be deceptive, such as when the target returns contain only interference that surpasses the detection threshold. Therefore, additional filters with clustering are applied, as described in Section 5.

4.3.2. MTI Algorithm

In this step, the MTI algorithm is applied to exclude static clutter points. This process necessitates the use of range and velocity information because it filters out static targets from the FoV and removes points corresponding to static targets (i.e., points that appeared in the previous frame). To remove these points, the static targets are mapped onto a vertical line that corresponds to a velocity of 0 m/s, and the Doppler channels associated with negligible velocities are removed from range-Doppler localization.

By adjusting the CFAR threshold, most non-target dynamic interference can also be removed. However, the dynamic interference may still be difficult to remove. This is because some distractors are moving at high or low speeds, whereas others are moving at speeds close to drones, such as when humans are walking. A threshold that is too small results in too much dynamic interference, whereas a threshold that is too large results in part of the drone and non-drone targets not being detected.

4.4. Angle Fourier Transform

The AOA estimation requires the use of at least two receiving antennas. The reflected signal from the target is received by both antennas; however, it must travel an additional distance β to reach the second antenna. Minor movement in the target location causes a phase shift across the receiving antennas. The phase difference between the two receiving antennas along the elevation φ_E and azimuth φ_A is determined as follows [21,43]:

$$\varphi_E = \frac{2\pi\beta \sin(\theta)}{\lambda}, \quad (8)$$

$$\varphi_A = \frac{2\pi\beta \cos(\theta)\sin(\varnothing)}{\lambda}, \quad (9)$$

where θ and $\varnothing$ are the elevation and azimuth angles of a reflecting target. β is the distance between two receiving antennas, and λ is the wavelength of the signal. Owing to slight differences in the phases of the received signals, θ and $\varnothing$ can be calculated as follows:

$$\theta = \sin^{-1}\left(\frac{\lambda\varphi_E}{2\pi\beta}\right), \quad (10)$$

$$\varnothing = \sin^{-1}\left(\frac{\cos(\theta)\lambda\varphi_E}{2\pi\beta}\right), \quad (11)$$

The angle-FFT is applied to the 2D range-Doppler localization, resulting in a 3D range-Doppler angle cube. Consider a point cloud $P = \{P_1, \dots, P_n\}$, where n is the number of detected targets. Each point P_i^t is represented by a feature set at a certain time t and is denoted by $P_i^t = [x, y, z, v, \varnothing, \theta]_i^t$, where x , y , and z are the 3D spatial coordinates.

5. Clustering

5.1. Cloud Point Clustering

The generated point clouds are sparse and insufficiently informative for recognizing distinct targets in the FoV. Furthermore, while static targets and noisy background reflections were removed through interference filtering, as discussed in Section 4, the remaining points are not always reflected by the targets. As shown in Figure 4, these interference points can be significant and can lead to confusion with points from nearby targets. Therefore, in this module, a clustering algorithm was applied to remove noise points in the point cloud, in addition to grouping sparse point clouds into several clusters, each corresponding to a single target present in the FoV.

The density-based spatial clustering of applications with noise (DBScan) algorithm was applied as a clustering algorithm [44], which is a density-aware clustering method that separates cloud points based on the Euclidean distance in 3D space. The DBScan algorithm groups several points in high-density regions into clusters, whereas interference points occur in low-density regions and are, therefore, removed from the clusters. In each

frame, DBScan scans all points sequentially, enlarging a cluster until a certain density connectivity criterion is no longer satisfied. Unlike K-means, DBScan does not require previous knowledge of the number of clusters and is, hence, well-suited for target detection problems with an arbitrary number of targets.

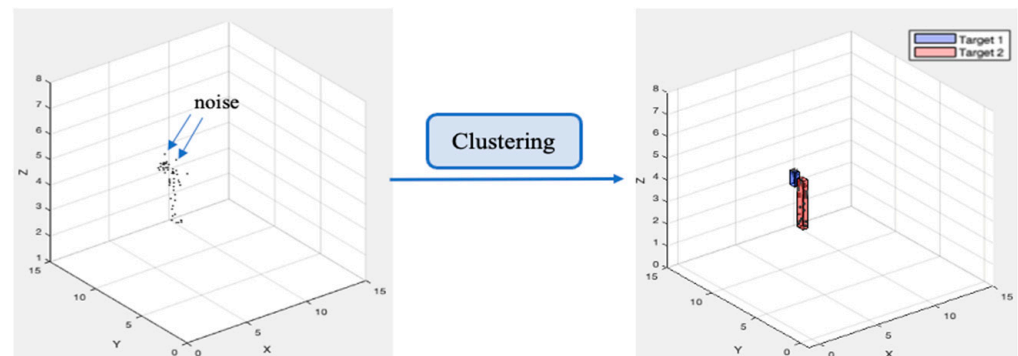


Figure 4. Clustering process input and output.

The distance between two points is used as the distance metric in DBScan for density-connection checking and is defined as follows:

$$d^t(i, j) = \alpha_x(x_i^t - x_j^t)^2 + \alpha_y(y_i^t - y_j^t)^2 + \alpha_z(z_i^t - z_j^t)^2 + \alpha_v(v_i^t - v_j^t)^2, \quad (12)$$

where $\alpha = [\alpha_x, \alpha_y, \alpha_z, \alpha_v]$ is the weight vector used to balance the contribution of each element. The parameters α_x , α_y , and α_z regulate the contribution of the distance between the two points in the x , y , and z axes, respectively. α_v regulates the contribution of the object speed. Velocity information is applied during the clustering phase to distinguish between two nearby targets with varying speeds, such as when two targets pass face-to-face. The clustering algorithm is illustrated in Algorithm 1.

Algorithm 1 Clustering algorithm.

Input: *maxDistance*: the largest Euclidean distance between two points; *minClusterPoint* : the minimum number of points required for a cluster.

Output: clustered targets

1. Initialize the values of *maxDistance* and *minClusterPoint*.
 2. Choose point p_i randomly, which is not identified as a cluster or noise.
 3. Calculate its neighboring points to determine whether it is a primary point. If this is true, form a cluster surrounding this point; otherwise, this point is considered noise.
 4. If p_i is a primary point, a cluster is formed, enlarging the cluster by including points that are reachable by it and are less than *maxDistance*.
 5. If a noise point is added, change the status of that point from noise to a boundary point.
 6. Continue with Steps 2:5 until all points have been designated as cluster points or noise.
-

5.2. Referencing

After clustering, each point is identified by the index of the cluster or noise point flag. For each clustered target, a reference point must be determined. To distinguish between different objects within the FoV, this reference point is used for tracking and retrieving the track information. In this paper, cluster centroid of each cluster is assigned to be the reference point. The algorithms can determine the centroid location of a cluster with a low misclassification rate, as illustrated in Algorithm 2.

Algorithm 2 Centroid-determining algorithm.**Input:** cloud point clusters.**Output:** centroid of each cluster

For each cluster:

1. Choose a point m as a centroid randomly.
2. Assign all remaining points as non-centroids represented by the nearest centroid.
3. Choose a non-centroid point m_{random} randomly selected in every cluster.
4. Let each current centroid denoted as m_i .
5. To form the new centroid, calculate the cost C of exchanging m_i with m_{random} , involving the cost of reassigning non-centroid points caused by the exchange if $C < 0$, and then, exchange m_i with m_{random} .
6. Repeat Step 3:5 until no change.

5.3. Cluster Box Estimation

All detected targets are enclosed in cluster boxes. The outermost points of each cluster are scanned and used to approximate the size of the 3D bounding box. The result of applying cluster box estimation at frame t is a collection of detected targets $O^t = [O_1, O_2, \dots, O_n]^t$, where n is the number of detected targets that might differ across frames. Each target O_i^t , is represented as a nine-dimensional vector comprising centroid 3D spatial coordinates x , y , and z and the length, height, and width of the 3D bounding box, l , h , and w . Specifically, the i -th point is denoted as $O_i^t = [x, y, z, v, \varnothing, \theta, l, h, w]^t$.

6. Tracking

During the tracking phase, the new position of the detected target is estimated sequentially, as shown in Figure 5A, followed by the temporal association of the new estimated track and target cluster to create a continuous target track, as shown in Figure 5B. The workflow of the proposed multiple-target-tracking-algorithm is shown in Figure 6. The components of the proposed target tracker are explored in detail below.

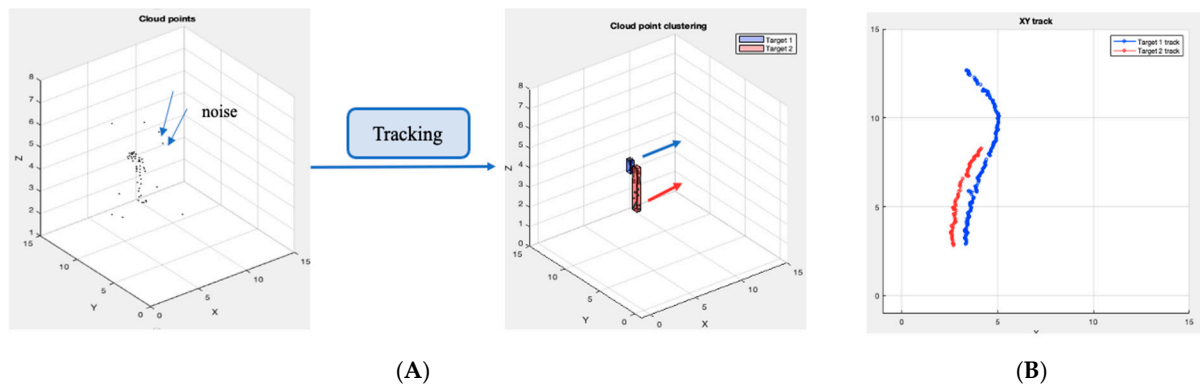


Figure 5. Tracking process input and output. (A) The estimation of the track at a certain frame. (B) The continuous track estimation.

6.1. Track Estimation and Updating

In the track estimation phase, the EKF [45,46] was adopted to predict the state tracks S_{t-1} to the current frame t , which is denoted as S_t^{est} . An EKF is a recursive linear filter used to determine the state of a dynamic system based on the time series of noisy observations. In addition, it features low computational complexity and a recursive structure and is resistant to measurement errors and correlations when dealing with multiple targets. Consequently, the radar community frequently uses a KF-based tracking technique [47]. Using the EKF when tracking a moving target will allow the system to detect the target even if it remains stationary, as well as to follow the target wherever it travels. In this paper, target tracking was performed utilizing distance and azimuth angle observations, rather

than radial velocity observations. Most investigations in the literature included radial velocity observations in the model, which caused the system to become too non-linear to produce meaningful estimates using KF. Moreover, observing only the distance limits the ability to locate a target in 3D space. This limitation can be overcome by observing the angular positions of moving targets and eventually reconstructing the full track of the target.

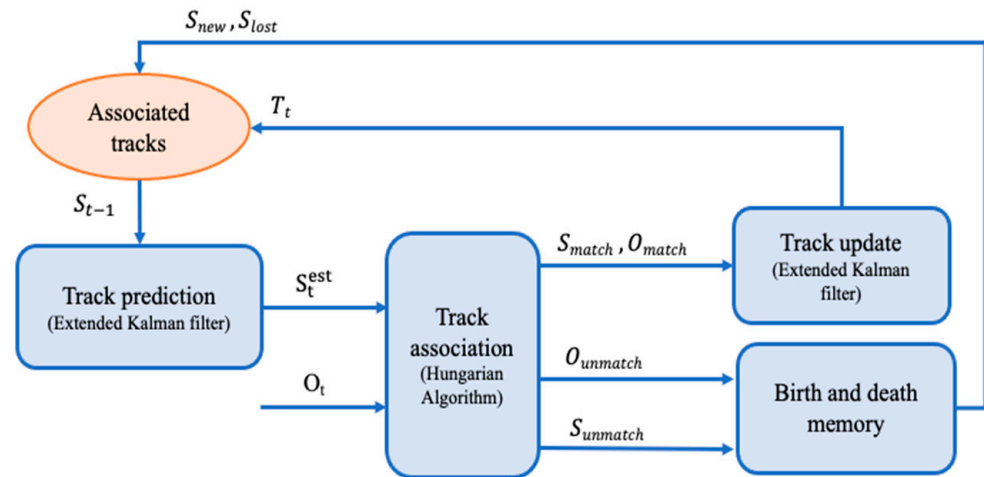


Figure 6. The workflow of the proposed target tracker.

Therefore, the KF model observation vector consists of the detected target's radial distance, and the azimuth angle at frame t is defined as $m^t = [r, \varnothing]^t$. A graphical representation of the radial distance and azimuth angle of the target from the ground station is shown in Figure 7. The current track state of the detected target at frame t is defined as $S^t := [x, y, z, v, \varnothing, \theta, l, h, w]^t$.

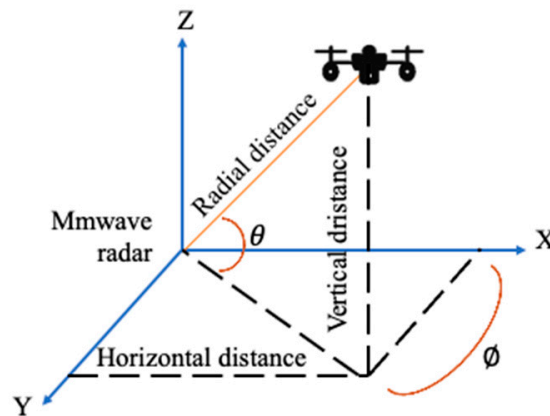


Figure 7. The radial distance and the angular position of the detected target.

The typical state-space representation of a non-linear time-discrete model is as follows:

$$S^t = AS^{t-1} + u^t, \quad (13)$$

$$M^t = h(T^t) + q^t, \quad (14)$$

where Equation (13) is responsible for explaining the evolution of the target states through t ; Equation (14) is responsible for matching the target's state to the measurements. u^t and q^t are the white Gaussian process noise and measurement noise, respectively.

$h(S^t) = \left[\sqrt{x^2 + y^2 + z^2}, \tan^{-1}\left(\frac{y}{x}\right) \right]^T$ represents the non-linear measurement process. A is the state transition matrix given in the time-discrete model and is defined as:

$$A = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (15)$$

To solve this non-linear measurement problem, a modified observation vector $\hat{M}_t = [r \cos \theta, r \sin \theta]$ is obtained using Equations (13) and (14). The EKF is used to estimate the new position of the detected targets in two steps. In the first step, the new track state is predicted by the mean S_{est}^t and covariance P_t at time t and is defined as:

$$S_{est}^t = AS^{t-1} + u^t, \quad (16)$$

$$P^T = AP^{t-1}A^T + q^t, \quad (17)$$

In the second step, the filter updates the first step state estimations using the Kalman gain K^t , which is denoted as:

$$K^t = P^t H^t T (H^t P^t H^t T + U)^{-1} \quad (18)$$

$$S^t = S^{t-1} + K^t (M^t - h(P^t)), \quad (19)$$

$$P^t = (I - K^t H^t) P^{t-1}, \quad (20)$$

where H^t is the Jacobian matrix of the partial derivatives of $h(\cdot)$, U is the noise covariance, and I is the identity matrix.

6.2. Track Association

Several successful single-target-tracking systems have been explored in the literature; however, tracking becomes difficult in the presence of multiple targets. The task of matching new tracks with target clusters from frame to frame in each input sequence has been proven to be complicated.

In the track association phase, the detected targets O_t and predicted track state S_t^{est} are associated at each frame. The Hungarian algorithm [48] was adopted to solve this many-to-many assignment problem with the objective of minimizing the combined distance loss. The procedure consists of two steps. In the first step, the actual cost matrix with a dimension of $O_t \times S_{t-1}$ is constructed using the squared Mahalanobis distance between the centroid of target detection O_t and the predicted track S_t^{est} for each frame t .

The cost matrix C_{ij}^{t} for the association between the predicted track i at $t - 1$ and the detected target j at the frame t is calculated as:

$$C_{ij}^t = (M_i^t - HS_i^t)^T (D^t)^{-1} (M_i^t - HS_i^t), \quad (21)$$

where $M_i^t - HS_i^t$ is the innovation process and D^t is the covariance matrix; it is calculated as $HP^tH + R$, and both are obtained as part of the KF update step.

The outputs of the track association module are a collection of detections $O_{match} = \{O_{match}^1, O_{match}^2, \dots, O_{match}^w\}$ matched with tracks $S_{match} = \{S_{match}^1, S_{match}^2, \dots, S_{match}^w\}$, along with the unmatched tracks $S_{unmatch} = \{S_{unmatch}^1, S_{unmatch}^2, \dots, S_{unmatch}^{m-w}\}$ and unmatched detected targets $O_{unmatch} = \{O_{unmatch}^1, O_{unmatch}^2, \dots, O_{unmatch}^{n-w}\}$, where w , m , and n are the number of matches, predicted tracks, and detected target, respectively.

6.3. Birth and Death

This module manages the newly appearing and disappearing tracks when the existing targets disappear and new tracks arise. In this paper, all unmatched detections $O_{unmatch}$ were considered as potential targets for entering the FoV. To avoid tracking false positives, a new track S_{new}^i is not created for $O_{unmatch}^i$ unless it has continually matched in the next few frames. However, all unmatched tracks are considered potential targets when leaving the FoV. To avoid deleting true positive tracks with a missing detection at specific frames, each unmatched track is kept for a few frames before being deleted.

7. Recognition

CNN Model for Feature Extraction and Classification

In this section, the proposed CNN model for multiple-target feature extraction and classification is presented for recognizing drone or non-drone targets. To overcome the non-uniformity in the number of points per frame and ensure a consistent length of input data, the data were processed before training and testing the CNN model. Regardless of the number of points in each frame, the point clouds in a 3D point cloud grid were converted into 2D occupancy grids. In this paper, we adopted the algorithm proposed in [49]. In particular, the cluster that encloses the points of a potential drone target was used to determine discriminative spatiotemporal patterns for each target individually.

However, a CNN model was periodically developed for spatiotemporal feature extraction throughout the 2D occupancy grids. To reduce network consumption and enhance training speed, the features of the point cloud data were directly used as the input data for the CNN, rather than mapping the point cloud to the images. Features extracted from the 2D occupancy grids of the cluster cloud points included the distance, velocity, and angle, as described in Section 4. Additional discriminative features were extracted, such as the height of the target and the size of the clusters.

The most-distinguishing features of drones are their ability to reach higher altitudes and their smaller sizes compared to pedestrians and other on-ground moving vehicles. The target altitude is determined by calculating the vertical distance as follows:

$$\lambda = r * \cos(\theta), \quad (22)$$

where λ is the height of the target from the ground.

The target size is determined by calculating the area of the clustered box as follows:

$$a = 2lw * 2lh * 2wh, \quad (23)$$

The CNN model consists of seven layers, as shown in Figure 8. Layer 1 is the input layer containing the six attributes: radial distance in meters, velocity in meters/second, azimuth angle in degrees, elevation angle in degrees, height in meters, and area in $meter^2$. Layer 2 is made up of six distinct modules, each of which is made up of a spatiotemporal convolution with kernel size 7×7 , maximum pooling composition with a sum size 3×3 , rectified linear unit (ReLU) activation functions, and maximum pooling with pooling area size 3×3 and a stride of two. Layer 3 is made up of six separated ResNet50. Layer 4 is made up of spatiotemporal convolution with kernel size 3×3 and average pooling with a pooling area of size 2×2 to fuse the six features. Layer 5 is made of an LSTM layer with an input size of 256 and a 128-cell hidden layer with a dropout probability of 0.5. Layer 6 is made of two fully connected (FC) layers and a ReLU activation function hidden layer to determine the output of the nodes in the FC. The final output layer would have three nodes, corresponding to the three classes of drone, non-drone, and drone and non-drone targets.

The CNN loss function is denoted as $Loss$ and is calculated as:

$$Loss(scr, t) = -scr[t] \log(\sum_j exp(scr[j])), \quad (24)$$

where scr is the classification score.

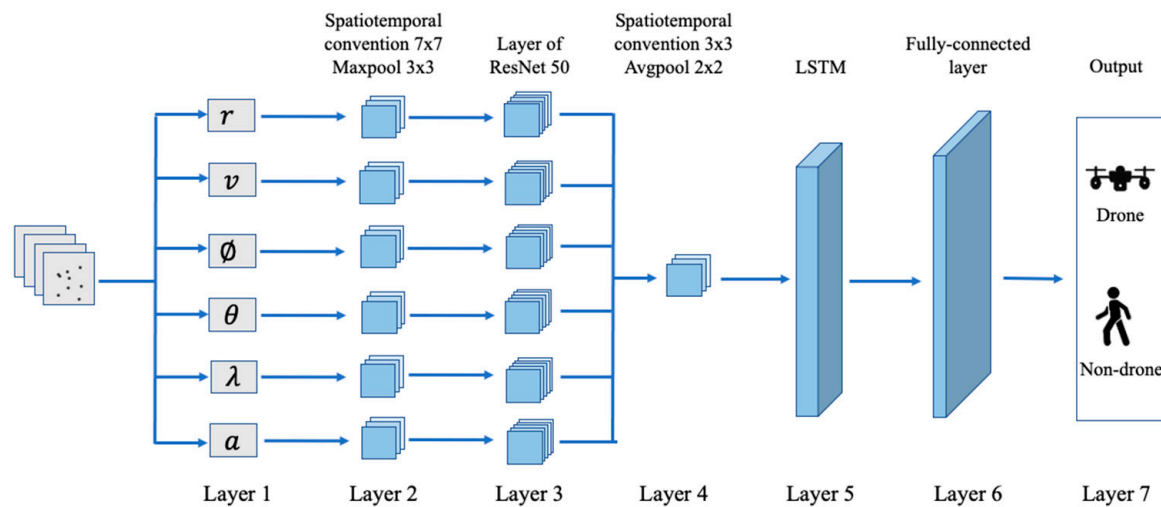


Figure 8. The proposed CNN model for drone target recognition.

The context flow of local and global time and space was designed using LSTM, which fuses the local and global spatiotemporal features. The six attributes of the point cloud served as the input independently, and spatiotemporal convolution kernels were applied to extract the spatiotemporal features of the point cloud. However, it is impossible for six distinct drone and non-drone target identification to adequately capture intrinsic features; hence, these six attributes must be fused. As a result, the fusion network is developed in Layer 4 to fuse the six features. After fusion, the features are more extensive and may more thoroughly describe a target's dimension, speed, altitude, and 3D spatial position coordinates.

8. Performance Evaluation

To evaluate the performance of the proposed framework, we used a dataset collected with an IWR6843ISK by Texas TI Single-chip 60 GHz to 64 GHz intelligent mmWave radar sensor [50] to train and test the CNN. There are several scenarios in the collected data. The drone and pedestrian data are mixed in Scenarios A, C, and D. Scenario B includes only noise and pedestrian data. In addition, the data were divided into training and validation sets at a 5:1 ratio. Our framework recognizes three classes of targets: drone, non-drone, and drone and non-drone, with a pedestrian as the non-drone target. In this context, the accuracy of the proposed algorithms for clustering, tracking, and recognition was determined as follows.

8.1. Clustering

The proposed clustering algorithm based on DBScan was compared to the well-known K-means clustering algorithm, as shown in Figure 9. The clustering accuracy achieved 88.2% for one target and 68.8% for two targets. The proposed clustering algorithm was more accurate than the K-means algorithm. To improve DBScan's performance, the weighting parameter α in Equation (12) must be defined. The large α causes the object to split into two clusters, whereas the small α causes loose clusters with many noise points. Practically, we discovered that $\alpha = 0.25$ yielded better clustering performance. Outliers were blended into clusters when $\alpha = 0$. Points related to a certain target were divided into two groups when $\alpha = 1$ (standard Euclidean distance).

8.2. Tracking

The estimated track accuracy is measured based on the RMSE in Figure 10. The RMSE value of the EKF for the target position was 0.21, whereas that of the LKF was 0.36. Based on the RMSE analysis, the EKF was more accurate than the LKF. The EKF and LKF are both methods used for dealing with 3D-radar-tracking systems. These filters aim to approximate

the non-linear functional model of a tracking system by using analytical techniques. In both the EKF and LKF, the non-linear equations of the model are approximated using a first-order expansion. This allows the use of the KF to estimate the state of the system after linearization. The main difference between the EKF and LKF lies in how they linearize the state space model. The EKF linearizes the model with respect to the estimated track, which is continuously updated using the collected information. On the other hand, the LKF linearizes the model with respect to a nominal track that has been pre-compiled without considering the collected information. It is important to note that the accuracy of the tracking performed by the LKF is heavily dependent on the accuracy of the nominal track that is predetermined. If the nominal track is inaccurate, it can lead to filter instability and poor tracking performance.

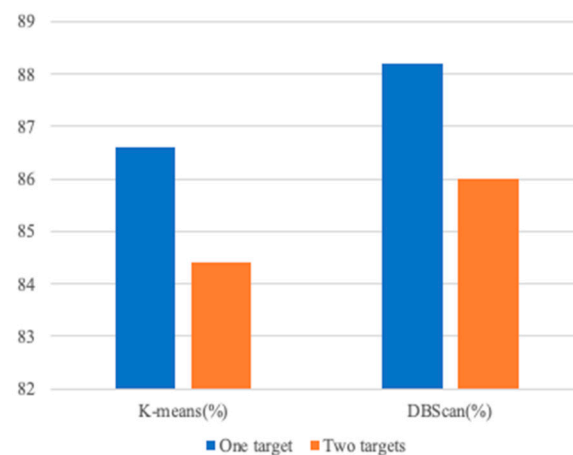


Figure 9. Clustering accuracy results.

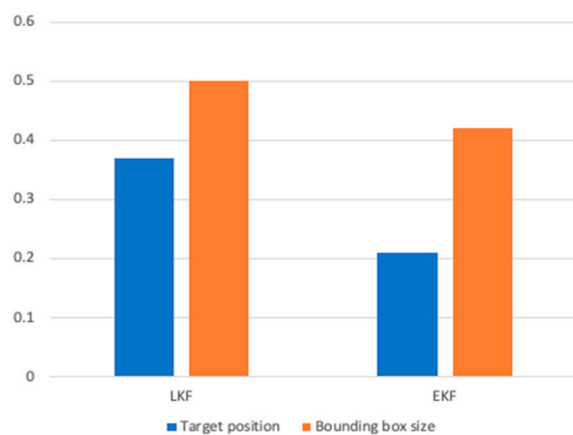


Figure 10. Tracking accuracy results.

Both the EKF and LKF are powerful tools for approximating non-linear models in radar-tracking systems, but they have different strategies for linearizing the models and estimating the state. It is important to carefully consider the trade-offs and limitations of each filter when deciding which to use in a specific application.

The RMSE for the target position is less than that for the bounding box size. The bounding box size is determined after cloud point clustering, which is used to extract the predicted track. In rare circumstances, the algorithm fails to extract the precise bounding box of the tracked object when noise points are not filtered from the FoV. In addition, EKF has a shorter computational time since transition matrices are not required for the calculation due to the linearization effect.

8.3. Recognition

To validate the proposed recognition model, the accuracies of two network modes, CNN and LSTM, were compared to the proposed CNN + LSTM model. CNN and LSTM comprise the first and second parts of the network, respectively. It can be observed from Figure 11 that the proposed model provided the best accuracy of 98.4% for one target and 98.1% for two targets.

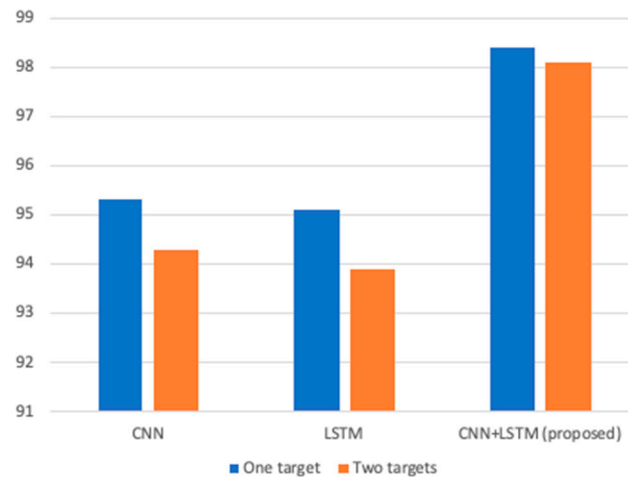


Figure 11. Recognition accuracy results.

9. Conclusions and Future Work

In this paper, we proposed a novel framework that performs simultaneous drone tracking and recognition using sparse cloud points generated from a low-cost small-sized mmWave radar sensor. Following detection, clustering, and Kalman filtering for location estimates in the 2D space plane, the raw data were processed further with a designed CNN classifier based on a cloud point spatiotemporal feature extractor. Our framework surpassed previous solutions in the literature in terms of a recognition accuracy of 98.4% for one target and 98.1% for two targets with a tracking RMSE of 0.21.

As part of future research, the proposed framework will be developed, and the dataset will be expanded to include drone-like targets such as birds.

Author Contributions: Conceptualization, S.S., E.A. and R.A.; methodology S.S.; software, S.S.; validation, S.S.; formal analysis, S.S., E.A. and R.A.; investigation, S.S.; resources, S.S.; data curation, S.S.; writing—S.S., E.A. and R.A.; writing—review and editing, S.S., E.A. and R.A.; visualization, S.S., E.A. and R.A.; supervision, S.S., E.A. and R.A.; project administration, S.S., E.A. and R.A.; funding acquisition, S.S., E.A. and R.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by The Deanship of Scientific Research at Taif University. The research number is 1-443-17.

Data Availability Statement: The data involved in the paper are available upon reasonable request to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kanellakis, C.; Nikolakopoulos, G. Survey on computer vision for UAVs: Current developments and trends. *J. Intell. Robot. Syst.* **2017**, *87*, 141–168.
2. Fu, R.; Al-Absi, M.A.; Kim, K.-H.; Lee, Y.-S.; Al-Absi, A.A.; Lee, H.-J. Deep Learning-Based Drone Classification Using Radar Cross Section Signatures at mmWave Frequencies. *IEEE Access* **2021**, *9*, 161431–161444.
3. Doole, M.; Ellerbroek, J.; Hoekstra, J. Estimation of traffic density from drone-based delivery in very low level urban airspace. *J. Air Transp. Manag.* **2020**, *88*, 101862.

4. Brahim, I.B.; Addouche, S.-A.; El Mhamedi, A.; Boujelbene, Y. Cluster-based WSA method to elicit expert knowledge for Bayesian reasoning—Case of parcel delivery with drone. *Expert Syst. Appl.* **2022**, *191*, 116160.
5. Sinhababu, N.; Pramanik, P.K.D. An Efficient Obstacle Detection Scheme for Low-Altitude UAVs Using Google Maps. In *Data Management, Analytics and Innovation*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 455–470.
6. Cheng, M.-L.; Matsuoka, M.; Liu, W.; Yamazaki, F. Near-real-time gradually expanding 3D land surface reconstruction in disaster areas by sequential drone imagery. *Autom. Constr.* **2022**, *135*, 104105.
7. Rizk, H.; Nishimur, Y.; Yamaguchi, H.; Higashino, T. Drone-Based Water Level Detection in Flood Disasters. *Int. J. Environ. Res. Public Health* **2022**, *19*, 237.
8. Nath, N.D.; Cheng, C.-S.; Behzadan, A.H. Drone mapping of damage information in GPS-Denied disaster sites. *Adv. Eng. Inform.* **2022**, *51*, 101450.
9. Mishra, B.; Garg, D.; Narang, P.; Mishra, V. Drone-surveillance for search and rescue in natural disaster. *Comput. Commun.* **2020**, *156*, 1–10. [[CrossRef](#)]
10. Jacob, B.; Kaushik, A.; Velavan, P. Autonomous Navigation of Drones Using Reinforcement Learning. In *Advances in Augmented Reality and Virtual Reality*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 159–176.
11. Chechushkov, I.V.; Ankusheva, P.S.; Ankushev, M.N.; Bazhenov, E.A.; Alaeva, I.P. Assessment of Excavated Volume and Labor Investment at the Novotemirsky Copper Ore Mining Site. In *Geoarchaeology and Archaeological Mineralogy*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 199–205.
12. Molnar, P. Territorial and Digital Borders and Migrant Vulnerability Under a Pandemic Crisis. In *Migration and Pandemics*; Springer: Cham, Switzerland, 2022; pp. 45–64.
13. Sharma, N.; Saqib, M.; Scully-Power, P.; Blumenstein, M. SharkSpotter: Shark Detection with Drones for Human Safety and Environmental Protection. In *Humanity Driven AI*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 223–237.
14. Bhatia, J.; Dayal, A.; Jha, A.; Vishvakarma, S.K.; Soumya, J.; Srinivas, M.B.; Yalavarthy, P.K.; Kumar, A.; Lalitha, V.; Koorapati, S. Object Classification Technique for mmWave FMCW Radars using Range-FFT Features. In Proceedings of the 2021 International Conference on COMMunication Systems & NETworks (COMSNETS), Bangalore, India, 5–9 January 2021; pp. 111–115.
15. Huang, X.; Cheena, H.; Thomas, A.; Tsoi, J.K.P. Indoor Detection and Tracking of People Using mmWave Sensor. *J. Sens.* **2021**, *2021*, 6657709.
16. Beringer, R.; Sixsmith, A.; Campo, M.; Brown, J.; McCloskey, R. The “acceptance” of ambient assisted living: Developing an alternate methodology to this limited research lens. In *Towards Useful Services for Elderly and People with Disabilities, Proceedings of the International Conference on Smart Homes and Health Telematics, Montreal, QC, Canada, 20–22 June 2011*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 161–167.
17. Ferris, D.D., Jr.; Currie, N.C. Microwave and millimeter-wave systems for wall penetration. In *Targets and Backgrounds: Characterization and Representation IV*; International Society for Optics and Photonics: Bellingham, WA, USA, 1998; Volume 3375, pp. 269–279.
18. Zhao, P.; Lu, C.X.; Wang, J.; Chen, C.; Wang, W.; Trigoni, N.; Markham, A. mID: Tracking and identifying people with millimeter wave radar. In Proceedings of the 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS), Santorini, Greece, 29–31 May 2019; pp. 33–40.
19. Sengupta, A.; Jin, F.; Zhang, R.; Cao, S. mm-Pose: Real-time human skeletal posture estimation using mmWave radars and CNNs. *IEEE Sens. J.* **2020**, *20*, 10032–10044.
20. Yang, Y.; Hou, C.; Lang, Y.; Yue, G.; He, Y.; Xiang, W. Person identification using micro-Doppler signatures of human motions and UWB radar. *IEEE Microw. Wirel. Compon. Lett.* **2019**, *29*, 366–368.
21. Tripathi, S.; Kang, B.; Dane, G.; Nguyen, T. Low-complexity object detection with deep convolutional neural network for embedded systems. In *Applications of Digital Image Processing XL*; International Society for Optics and Photonics: Bellingham, WA, USA, 2017; Volume 10396, p. 103961M.
22. Wang, S.; Song, J.; Lien, J.; Poupyrev, I.; Hilliges, O. Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum. In Proceedings of the 29th Annual Symposium on User Interface Software and Technology, Tokyo, Japan, 16–19 October 2016; pp. 851–860.
23. Palipana, S.; Salami, D.; Leiva, L.A.; Sigg, S. Pantomime: Mid-air gesture recognition with sparse millimeter-wave radar point clouds. In Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, New York, NY, USA, 21–26 September 2021; Volume 5, pp. 1–27.
24. Srigrarom, S.; Sie, N.J.L.; Cheng, H.; Chew, K.H.; Lee, M.; Ratsamee, P. Multi-camera Multi-drone Detection, Tracking and Localization with Trajectory-based Re-identification. In Proceedings of the 2021 Second International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP), Bangkok, Thailand, 20–22 January 2021; pp. 1–6.
25. Al-Emadi, S.; Al-Ali, A.; Al-Ali, A. Audio-Based Drone Detection and Identification Using Deep Learning Techniques with Dataset Enhancement through Generative Adversarial Networks. *Sensors* **2021**, *21*, 4953. [[PubMed](#)]
26. Alsoliman, A.; Rigoni, G.; Levorato, M.; Pinotti, C.; Tippenhauer, N.O.; Conti, M. COTS Drone Detection using Video Streaming Characteristics. In Proceedings of the 2021 International Conference on Distributed Computing and Networking, Nara, Japan, 5–8 January 2021; pp. 166–175.
27. Rzewuski, S.; Kulpa, K.; Pachwicewicz, M.; Malanowski, M.; Salski, B. *Drone Detectability Feasibility Study using Passive Radars Operating in WIFI and DVB-T Band*; The Institute of Electronic Systems: Warsaw, PL, USA, 2021.

28. Svanström, F.; Englund, C.; Alonso-Fernandez, F. Real-Time Drone Detection and Tracking with Visible, Thermal and Acoustic Sensors. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 7265–7272.
29. Yoo, L.-S.; Lee, J.-H.; Lee, Y.-K.; Jung, S.-K.; Choi, Y. Application of a Drone Magnetometer System to Military Mine Detection in the Demilitarized Zone. *Sensors* **2021**, *21*, 3175.
30. Dogru, S.; Baptista, R.; Marques, L. Tracking drones with drones using millimeter wave radar. In *Advances in Intelligent Systems and Computing, Proceedings of the Robot 2019: Fourth Iberian Robotics Conference, Porto, Portugal, 20–22 November 2019*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 392–402.
31. Semkin, V.; Yin, M.; Hu, Y.; Mezzavilla, M.; Rangan, S. Drone detection and classification based on radar cross section signatures. In Proceedings of the 2020 International Symposium on Antennas and Propagation (ISAP), Osaka, Japan, 25–28 January 2021; pp. 223–224.
32. Rai, P.K.; Idsøe, H.; Yakkati, R.R.; Kumar, A.; Khan, M.Z.A.; Yalavarthy, P.K.; Cenkeramaddi, L.R. Localization and Activity Classification of Unmanned Aerial Vehicle using mmWave FMCW Radars. *IEEE Sens. J.* **2021**, *21*, 16043–16053.
33. Rahman, S.; Robertson, D.A. FAROS-E: A compact and low-cost millimeter wave surveillance radar for real time drone detection and classification. In Proceedings of the 2021 21st International Radar Symposium (IRS), Berlin, Germany, 21–22 June 2021; pp. 1–6.
34. Jokanovic, B.; Amin, M.G.; Ahmad, F. Effect of data representations on deep learning in fall detection. In Proceedings of the 2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM), Rio de Janeiro, Brazil, 10–13 July 2016; pp. 1–5.
35. Li, J.; Yang, B.; Chen, C.; Habib, A. NRLI-UAV: Non-rigid registration of sequential raw laser scans and images for low-cost UAV LiDAR point cloud quality improvement. *ISPRS J. Photogramm. Remote Sens.* **2019**, *158*, 123–145.
36. Molchanov, P.; Gupta, S.; Kim, K.; Pulli, K. Short-range FMCW monopulse radar for hand-gesture sensing. In Proceedings of the 2015 IEEE Radar Conference (RadarCon), Arlington, VA, USA, 10–15 May 2015; pp. 1491–1496.
37. Janakaraj, P.; Jakkala, K.; Bhuyan, A.; Sun, Z.; Wang, P.; Lee, M. STAR: Simultaneous tracking and recognition through millimeter waves and deep learning. In Proceedings of the 2019 12th IFIP Wireless and Mobile Networking Conference (WMNC), Paris, France, 11–13 September 2019; pp. 211–218.
38. Mafukidze, H.D.; Mishra, A.K.; Pidanic, J.; Francois, S.W.P. Scattering Centers to Point Clouds: A Review of mmWave Radars for Non-Radar-Engineers. *IEEE Access* **2022**, *10*, 110992–111021.
39. Shuai, X.; Shen, Y.; Tang, Y.; Shi, S.; Ji, L.; Xing, G. millieye: A lightweight mmwave radar and camera fusion system for robust object detection. In Proceedings of the International Conference on Internet-of-Things Design and Implementation, Charlottesville, VA, USA, 18–21 May 2021; pp. 145–157.
40. Scharf, L.L.; Demeure, C. *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*; Addison-Wesley: Reading, MA, USA, 1991.
41. Ash, M.; Ritchie, M.; Chetty, K. On the application of digital moving target indication techniques to short-range FMCW radar data. *IEEE Sens. J.* **2018**, *18*, 4167–4175.
42. Richards, M.; Holm, W.; Scheer, J. Principles of Modern Radar: Basic Principles, ser. *Electromagn. Radar Inst. Eng. Technol.* **2010**, *6*, 47.
43. Pegoraro, J.; Rossi, M. Real-time people tracking and identification from sparse mm-wave radar point-clouds. *IEEE Access* **2021**, *9*, 78504–78520.
44. Birant, D.; Kut, A. ST-DBSCAN: An algorithm for clustering spatial-temporal data. *Data Knowl. Eng.* **2007**, *60*, 208–221.
45. Kalman, R.E. A new approach to linear filtering and prediction problems. *J. Basic Eng. Mar.* **1960**, *82*, 35–45.
46. Fujii, K. Extended kalman filter. *Ref. Man.* **2013**, *14*, 14–22.
47. Blackman, S.S. *Multiple-Target Tracking with Radar Applications*; Artech House, Inc.: Dedham, MA, USA, 1986.
48. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97.
49. Huesman, J. Converting 3D point cloud data into 2D occupancy grids suitable for robot applications. In *NDSU EXPLORE: Undergraduate Excellence in Research and Scholarly Activity*; North Dakota State University: Fargo, ND, USA, 2015.
50. Texas Instruments. *Single-Chip 60-GHz to 64-GHz Intelligent mmWave Sensor Integrating Processing Capability*; Texas Instruments: Dallas, TX, USA, 2022.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.