

## Article

# RF Signal-Based UAV Detection and Mode Classification: A Joint Feature Engineering Generator and Multi-Channel Deep Neural Network Approach

Shubo Yang<sup>1,2</sup>, Yang Luo<sup>3,\*</sup> , Wang Miao<sup>4</sup>, Changhao Ge<sup>5</sup>, Wenjian Sun<sup>3</sup> and Chunbo Luo<sup>2,3,\*</sup>

<sup>1</sup> Glasgow College, University of Electronic Science and Technology of China, Chengdu 611731, China; 2018190607005@std.uestc.edu.cn

<sup>2</sup> Yangtze Delta Region Institute (Huzhou), University of Electronic Science and Technology of China, Huzhou 313001, China

<sup>3</sup> School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China; sunwenjian@std.uestc.edu.cn

<sup>4</sup> Department of Computer Science, University of Exeter, Exeter EX4 4QF, UK; wang.miao@exeter.ac.uk

<sup>5</sup> James Watt School of Engineering, University of Glasgow, Glasgow G12 8QQ, UK; 2357902G@student.gla.ac.uk

\* Correspondence: luoyang@uestc.edu.cn (Y.L.); c.luo@uestc.edu.cn (C.L.)



**Citation:** Yang, S.; Luo, Y.; Miao, W.; Ge, C.; Sun, W.; Luo, C. RF Signal-Based UAV Detection and Mode Classification: A Joint Feature Engineering Generator and Multi-Channel Deep Neural Network Approach. *Entropy* **2021**, *23*, 1678. <https://doi.org/10.3390/e23121678>

Academic Editors: Andrea Prati, Carlos A. Iglesias, Luis Javier García Villalba and Vincent A. Cicirello

Received: 18 November 2021

Accepted: 9 December 2021

Published: 14 December 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 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/>).

**Abstract:** With the proliferation of Unmanned Aerial Vehicles (UAVs) to provide diverse critical services, such as surveillance, disaster management, and medicine delivery, the accurate detection of these small devices and the efficient classification of their flight modes are of paramount importance to guarantee their safe operation in our sky. Among the existing approaches, Radio Frequency (RF) based methods are less affected by complex environmental factors. The similarities between UAV RF signals and the diversity of frequency components make accurate detection and classification a particularly difficult task. To bridge this gap, we propose a joint Feature Engineering Generator (FEG) and Multi-Channel Deep Neural Network (MC-DNN) approach. Specifically, in FEG, data truncation and normalization separate different frequency components, the moving average filter reduces the outliers in the RF signal, and the concatenation fully exploits the details of the dataset. In addition, the multi-channel input in MC-DNN separates multiple frequency components and reduces the interference between them. A novel dataset that contains ten categories of RF signals from three types of UAVs is used to verify the effectiveness. Experiments show that the proposed method outperforms the state-of-the-art UAV detection and classification approaches in terms of 98.4% and F1 score of 98.3%.

**Keywords:** unmanned aerial vehicles; UAV detection; UAV mode classification; Feature Engineering Generator; multi-channel deep neural network

## 1. Introduction

Unmanned aerial vehicles (UAVs), also called drones, are gaining increasing popularity since they have high flexibility, ease of affordability, and exceptional capability. The recent advances in UAV technology have led to the proliferation of aerial services in our sky, e.g., emergency networks [1], healthcare system [2], surveillance system [3], coastal engineering [4], transportation engineering [5], assistance of distressed people [6], and many more [7]. Meanwhile, UAVs are employed to improve wireless communication, because they can provide connections between devices [8], increase the energy efficiency through trajectory optimization [9], assist resource allocation [10], and set up relay links [11]. Apart from single UAV's usage, there is extensive research in applications of multiple UAVs, such as in radio navigation aids [12] and cellular networks [13]. The problems in communication

networks between multiple UAVs before realizations of stable and reliable context-specific networks are overviewed [14].

Since the UAVs have flexibility, ease of affordability, and controllability, they may be utilized for malicious purposes and thereby pose potential security and privacy threats [15,16]. The threats contain eavesdropping, invading restricted regions, attacks on infrastructure, and colliding with people after losing control. To address these issues, efficient air traffic management becomes indispensable to ensure the safety of UAV flight and management [17], which is critical to both themselves and the flying environment, and property managers need to be aware of an approaching UAV. Hence, as the foundation for the following-up regulating measures, the methods for UAV detection and mode classification are urgently required.

Conventional detection methods, e.g., radars, acoustics, and vision, are constrained under some conditions. For example, the radar-based detection methods are restricted by the UAV sizes, the image-based methods have limits of distinguishing birds from drones, and acoustics-based methods are usually influenced by noise and have short detection distance [18]. Different from the conventional methods, RF signals can be detected at long distances and are less influenced by environmental factors. There is intensive research into classifying UAV types by RF signals. The work in [19] first detected the UAV signals, and then used neighborhood component analysis (NCA) and machine learning classifiers for classification of 15 UAV controllers types. Moreover, based on a Native Bayes approach and features of energy transient signal inputted into machine learning algorithms, 14 types of UAV RF signals are classified [20]. Bhattacharjee et al. in [21] utilized a keysight sensor to detect the UAV type by comparing the received RF signature with other UAVs' RF signatures in a database. However, most existing works focus on detecting UAV types instead of UAV flight modes, which contain the information of UAVs' operation status and are significant for UAV safety management. Therefore, in this paper, we proposed an effective RF signal-based method to not only detect UAVs' types but also classify their flight modes.

The challenges of RF signal-based approaches are the similarities contained in the signals and the features of different frequency components. To tackle the challenges, our method first uses a Feature Engineering Generator (FEG) to extract features from RF signals. With the data preprocessed by FEG, we further design a DNN and a multi-channel deep neural network (MC-DNN) to classify the flight modes of UAVs. The multi-channel design separates different frequency components and reduces their corresponding interferences. The effectiveness of the method is verified based on a practical dataset in [22], where up to 10 categories of RF signals are included. The experiment results show that the proposed approach achieves an accuracy of 98.4% and an F1 score of 98.3%, and outperforms other state-of-the-art methods [22–24]. The main contributions of our work are summarized as follows.

- We design a joint FEG and MC-DNN approach for UAV detection and mode classification. The RF signals are preprocessed by FEG and then input into an MC-DNN for classification.
- In FEG, data truncation and normalization separates different components, the moving average filter removes the noise in the signals, and the concatenation exploits comprehensive details of the RF samples.
- We design MC-DNN to classify the signals preprocessed by the proposed FEG. The multi-channel input separates different frequency components of data to reduce interferences, and MC-DNN learns the classification effectively.
- We verify the joint approach through extensive experiments on an open dataset consisting of ten RF signal categories from three types of UAVs. Our method achieves high accuracy and F1 score and outperforms other methods.

The rest of the paper is organized as follows. Section 2 summarizes the related work. Section 3 describes the system model and problems. Section 4 presents the RF preprocessing

and DNN structure. Section 5 provides the experimental results of our method. Finally, Section 6 draws some conclusions.

## 2. Related Works

Radio detection and ranging (RADAR) uses electromagnetic waves to collect information of flying objectives, such as distance and angle [25]. Thus, radars as active sensors are adopted for UAV detection, tracking, and classification. Most radar systems are designed based on Doppler signatures [25,26] and phase-interferometry [27]. Moreover, the work in [28] classifies two scenarios by not only using the micro-Doppler signature but also the cyclostationarity signature of the UAV signals and pseudo-Doppler principle. Although some progress has been made in the area of radar-based UAV detection, the relatively small radar cross-sections make UAVs invisible to radars and barriers influence the propagation of radar signals.

Acoustic sensors are applied for UAV detection. Data mining techniques are used based on acoustics sampling data, where the Hidden Markov Model was applied to analyze the emitted sound of UAVs [29]. A low-cost acoustic array of dynamically placed microphones was adopted to locate far-field small UAVs using a delay-and-sum beamforming algorithm [30]. However, the acoustic-based methods are influenced by high background sound and limited by the operating distance.

There are also vision-based approaches for UAV detection. A vision system based on a standard RGB digital camera to track a known UAV and assist automatic landing was presented in [31]. A method for a UAV to detect and track a cooperative flying vehicle was proposed based on template matching and morphological filtering [32]. The work in [33] constructed a YOLOv3 object detector to extract features from images using computer vision and convolutional neural network (CNN). Although being a promising technology, vision-based methods are sensitive to blurring images and line of sight limitations, such as cloud and fog, making the methods challenging to be used in real-world scenarios.

Different from the abovementioned methods, methods based on RF signals can be applied in the real world more easily, being less constrained by UAV shapes and the uncertainties in the acquisition environment. Meanwhile, the UAV RF signals can be captured at a long distance and contain abundant information about the UAVs' flight modes [22,34], which cannot be easily achieved by other methods.

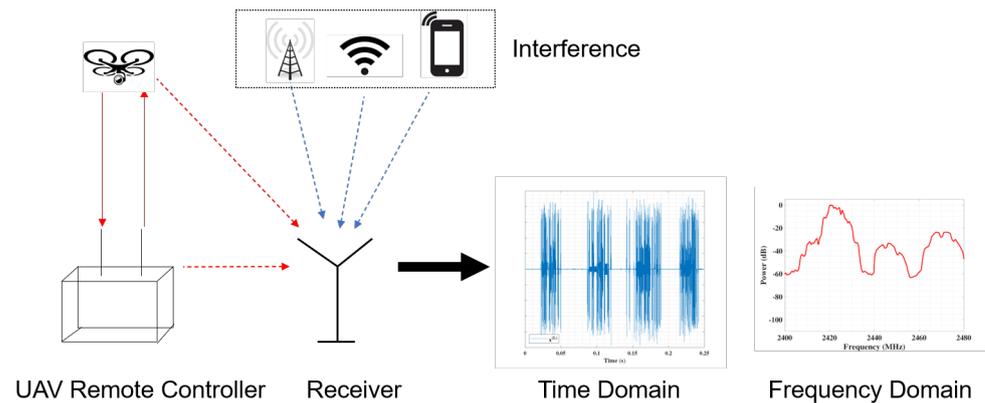
Since RF signals usually have a large amount of data, machine learning methods can be used to classify the RF signals. The neural networks in machine learning adapt the complex matches between the inputs and outputs of systems and are applied in many areas, such as speech recognition [35,36], human pose estimation [37], and image classification [38,39]. Neural networks automatically choose factors of the input to learn rather than relying on features picked by humans, which allows the methods to learn features more comprehensively and without biases. Therefore, using neural networks to classify UAV RF signals attracts considerable research efforts. The authors in [40] used wavelet transform analytics to extract unique signatures from the transient and steady state of the RF signals. A pretrained CNN-based model (SqueezeNet) was used to distinguish UAVs from interference and identify UAV types. The work in [41] trained CNNs using RF time-series images and spectrograms to classify 15 different drone controllers. The CNN model based on spectrograms was further applied denoising mechanism and was tested under different Signal-to-noise ratio (SNR) levels. However, the aforementioned research focused on the identification of UAV types from existing noise and interference instead of the UAV flight mode classification. Therefore, combining the advantages of RF signals and neural networks, we proposed a joint FEG and MC-DNN approach to not only detect UAV presence but also classify the UAV flight modes. We reduce the similarities and exploit the characteristics contained in different frequency components. Our method achieves high classification accuracy and F1 score, and our method outperforms other methods.

### 3. System Model and Problems

In this section, the system model is first introduced including RF signal acquisition, noise and interference, and RF signals in the frequency domain. Then, the challenges in UAV classification are presented as the problems to be tackled.

#### 3.1. System Model

We introduce the system model as in Figure 1 with a focus on the UAV operation signals' characteristics, which serve as the basis for further proposed classification method.



**Figure 1.** The system model of UAV RF signal acquisition.

##### 3.1.1. RF Signal Acquisition

There are mainly two kinds of RF signals generated between UAVs and their controllers: the uplink and the downlink. The uplink signal contains the controlling RF signal commands from the controller to UAVs, while the downlink one contains telemetry signals and video signals from UAVs to controllers. Most RF signals generated uniquely characterize UAVs due to the UAVs' circuitry design and modulation techniques [19]. Besides, most UAVs are operated at frequencies around 2.4 GHz [42]. By passively and continuously listening to the communication between the UAVs and controllers, the unique RF signals for different types of UAVs with different flight modes can be collected. Herein, tools such as universal software radio peripheral (USRP) can be used for signal acquisition, and the sampling rates are set to be larger than the Nyquist rate to avoid aliasing.

Capturing signals of a larger bandwidth gives comprehensive information of different frequency components. However, the devices can have bandwidth constraints to capture the RF signals. Thus, the whole bandwidth is divided into low-frequency and high-frequency components, where each component is captured by respective device.

##### 3.1.2. Noise and Interference in RF Signals

The interference results from other wireless sources that are also operated in the same UAVs operating frequency band, such as Wi-Fi and Bluetooth. Besides, the signal to interference plus noise ratio (SINR) of the captured RF signals is related to the upper limit of classification. Thus, the captured signals with UAVs are modeled as the combination of UAV RF signals and the background signals with noise and interference, captured from the ambient environment without operating UAVs.

##### 3.1.3. RF Signals in Frequency Domain

The RF signals are typically captured in the time domain, but the signals in the frequency domain have latent characteristics. Besides, directly using time-domain signals for classification has some drawbacks: First, the time-domain signals usually have a large size of data, which requires high computation resources for preprocessing. The conversion to the frequency domain significantly reduces the data size. Second, if the time-domain signals are further divided into segments, the start and end of the segments are randomly

chosen. This may result in a large portion of noise in some segments, and there are not enough features to conduct accurate UAV classifications. Third, many devices used to capture signals have bandwidth limits. The direct concatenation of bandwidth-limited time-domain signals is questionable while the frequency domain spectra are not. To avoid the abovementioned drawbacks and reveal more features, the RF signals are converted into the frequency domain by Fourier Transform.

### 3.2. Problems

The RF signal dataset includes the background RF signal and the RF signals of different UAVs with diverse flight modes. The UAV presence must be first classified. Consequently, the types of UAVs and flight modes of UAVs should be classified. The challenges lie in mainly four aspects: the similarities between distinct types of UAV signals, the similarities between the same type of UAVs with different flight modes, the existence of noise and interference, and distinguishing diverse features of different frequency components. This paper focuses on proposing a novel joint FEG and MC-DNN approach to overcome these challenges and classify the UAV flight modes accurately.

## 4. Methodology

To solve the problems, the proposed method consists of FEG preprocessing and MC-DNN. The method concentrates on boosting the discrepancies between each category of RF signals, separating different features represented by each frequency component, and learning representative features. Specifically, the FEG extracts more distinguishable features and reduces the influence of biases. The MC-DNN automatically selects features from preprocessed signals and learns the relationship between the input and the objectives effectively.

### 4.1. Feature Engineering Generator

The objective of feature engineering is to reveal features from raw data since the features represent the data better, the more accurate performance obtained. Thus, the FEG aims at separating different frequency components and reducing the similarities of signals. FEG uses three techniques: data truncation and normalization, moving average filter, and concatenation.

#### 4.1.1. Data Truncation and Normalization

The dataset of RF signal captured in the frequency domain is composed of low-frequency and high-frequency components. The components may own different features and exhibit different power levels. Thus, normalizing two components together leads to that the frequency component with small values is dominated by the other one. This means that the small values are normalized to nearly zero, and the value changes become almost invisible. To address this issue, we truncate the two components into two sub-datasets. Each sub-dataset is normalized individually to fully extract different features.

#### 4.1.2. Moving Average Filter

Since the existence of noise and the frequency spectra after Fourier Transform have oscillations, a  $n$ -point moving average filter is proposed to smooth the spectra and reduce the noise effects. While the noise is random, the UAV signals remain almost unchanged. The noise adds destructively in the filter, and the oscillations are reduced while keeping the substantial trend of UAV RF signals. The moving average filter is calculated as

$$\bar{p} = \frac{1}{n} \sum_{i=0}^{n-1} p_i, \quad (1)$$

where  $p_i$  is the input value,  $\bar{p}$  is the output value, and  $n$  is the number of inputs. The output is the mean of adjacent  $n$  values. Due to different noise of the frequency components, the

parameter  $n$  in Equation (1) for each component is chosen separately. When the frequency signals in the component have larger oscillations, bigger  $n$  is expected and more samples are averaged.

#### 4.1.3. Concatenation

Using individually normalized sub-dataset for classification can't fully exploit the complete details of the dataset. Hence, the sub-datasets are concatenated to provide a comprehensive view of RF samples. Some concatenation ways alter the sub-datasets and reduce features, such as multiplying coefficients with the first samples of the high-frequency component [22]. Multiplication achieves continuity between the two components, but it results in diminishing small values and changing values. Direct concatenation connects the components without modification and keeps distinct respective features. Besides, the continuity between the low-frequency and high-frequency components is not necessary for classification.

In conclusion, the overall FEG algorithm is presented in Algorithm 1. The RF data is truncated into low-frequency and high-frequency components, resegmented, operated by Fourier Transform, moving average filtered separately, concatenated together, and labeled.

---

#### Algorithm 1 Feature Engineering Generator Algorithm.

---

##### Require:

The original low-frequency time domain component  $L$ .

The original high-frequency time domain component  $H$ .

The number of samples in each data segment  $M$ .

The number of categories of data from different types of UAVs with flight modes  $N$ .

The points of moving average filter for the low-frequency and high-frequency components  $n_l$  and  $n_h$ , respectively.

##### Ensure:

The Feature Engineering Generator preprocessed frequency domain data  $D$ .

- 1: **for**  $n$  in  $N$  **do**
  - 2:   Extract the time domain low-frequency component  $L_n$  and high-frequency component  $H_n$  of category  $n$ .
  - 3:   Resegment  $L_n, H_n$  into new segments with  $M$  samples per segment  $S_L, S_H$ , respectively.
  - 4:   **for**  $l$  in  $S_L$  **do**
  - 5:     Fourier transform  $l$ .
  - 6:      $n_l$ -point moving average filter  $l$ .
  - 7:   **end for**  $l$
  - 8:   **for**  $h$  in  $S_H$  **do**
  - 9:     Fourier transform  $h$ .
  - 10:     $n_h$ -point moving average filter  $h$ .
  - 11:   **end for**  $h$
  - 12:    $S_L = \frac{S_L}{\max(S_L)}; S_H = \frac{S_H}{\max(S_H)}$ .
  - 13:    $S = (S_L; S_H)$ .
  - 14:    $D_n = (S^2; n)$ .
  - 15: **end for**  $n$
  - 16:  $D = (D_1, D_2, \dots, D_n)$ .
- 

#### 4.2. DNN Structure

Given the RF signals preprocessed by FEG, DNNs are designed to solve the multi-class classification problem. DNNs can automatically select and learn the features in the RF signals. A well-designed DNN is capable of adapting the relationship between the input and objective. In this section, a DNN structure is first designed for classification. The performance of the DNN is also contrasted as a baseline. Next, the multi-channel technique is applied and a multi-channel input DNN is designed for better classification performance.

### 4.2.1. Deep Neural Network

Based on feedforward artificial neural networks called multilayer perceptron, a DNN in Figure 2 is designed to classify the RF signals, which includes the model input and output, DNN structure, and loss function.

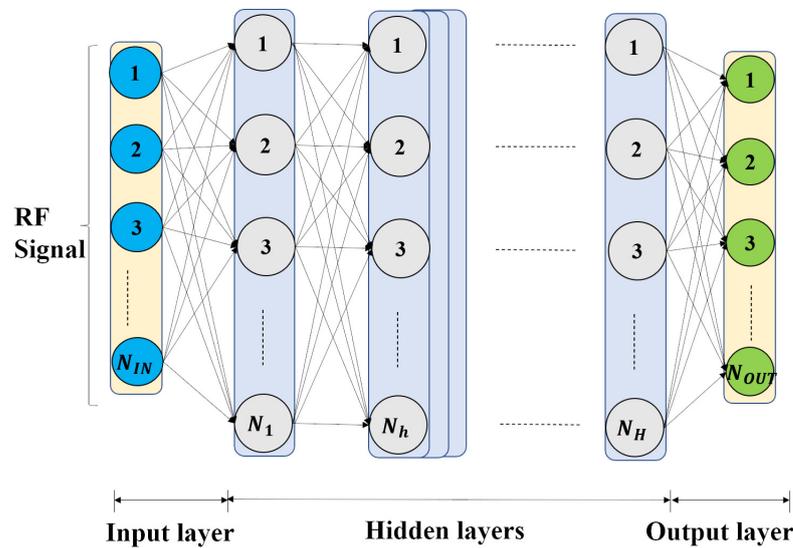


Figure 2. The structure of the DNN with an input layer,  $H$  hidden layers and an output layer.

#### Input and Objective

The DNN classifies the RF signal data, and the classes are encoded by one-hot encoding into sequences of numbers with 1 representing the corresponding class and 0 representing other classes. Each objective for one piece of input data is a vector, and its dimension is the number of classes.

#### Deep Neural Network Structure

The DNN has  $H$  hidden layers with  $N_h$  neurons in layer  $h$ , and the structure details are explained in [43]. The leftmost layer is the input layer with  $N_{IN}$  neurons, being equal to the dimension of input RF signals preprocessed by FEG. The rightmost layer is the output layer with  $N_{OUT}$  neurons, being equal to the number of classes. Each layer receives all the outputs of the previous layer and operates the calculation as follows.

$$\begin{aligned} z_l &= W_l^T a_{l-1} + b_l, \\ a_l &= \delta_l(z_l), \end{aligned} \tag{2}$$

where  $a_l$  is the output vector of layer  $l$ ,  $a_{l-1}$  is the output vector of the previous layer,  $W_l$  is the weight vector,  $b_l$  is the bias vector, and  $\delta_l(\cdot)$  is an activation function, e.g., rectified linear unit function (ReLU) and Softmax function. The weights and biases of each layer are determined through a supervised learning process. A loss function is minimized by a gradient descent algorithm.

#### Loss Function

The DNN's loss function  $L$  is defined as the mean square error between the outputs and the objectives as follows.

$$L(d_i, \hat{d}_i) = \frac{1}{C} \sum_{c=1}^C (d_i(c) - \hat{d}_i(c))^2, \tag{3}$$

where  $d_i$  is a vector of the objectives,  $\hat{d}_i$  is a vector of the final layer outputs, and  $C$  is the total number of outputs. The objective of the DNN is to minimize the loss function. During this process, the DNN learns the relationship and improves accuracy.

#### Stratified K-Fold Cross-Validation

To estimate the performance and effectiveness of the DNN on a limited dataset, stratified K-fold cross-validation [44] is adopted. The signals and objectives are shuffled randomly and divided into  $K$  folds evenly. The number of samples per category in each fold is proportional to the category's portion in the dataset. There are  $K$  training and testing cycles, where  $K - 1$  folds are the training set and the remaining fold is for testing. Each fold is used to test once and train  $K - 1$  times. Training on the same training set can lead to overfitting on the training set and perform badly on unseen data. The overall performance metrics are summarized by taking the mean of  $K$  results. The cross-validation average result provides a steady evaluation and objectively reflects the performance of a network.

#### Confusion Matrix

The confusion matrix, or error matrix, is used to evaluate the performance of a classifier [45] by giving details into the errors and their types. It visualizes the overall accuracy by comparing the actual objectives and predicted classes. The columns of the confusion matrix represent the output class, while the rows represent the predicted classes. Several performance metrics are specified in the confusion matrix, e.g., recall, precision, false discovery rate (FDR), false-negative rate (FNR), accuracy, error, and F1 score.

#### 4.2.2. Multi-Channel DNN

Various factors contribute to the final classification result and the factors have little correlation with each other. Here, the low-frequency and high-frequency components have respective features and relationships to the objective. The multi-channel input technique enables the model to consider more possible factors and prevent factors from interfering with each other. Hence, based on DNN, multi-channel DNN (MC-DNN) is designed. Different from the first hidden layer following two components in DNN, the two FEG preprocessed components are input separately to the MC-DNN in Figure 3. The first channel input is the low-frequency component and the second channel input is the high-frequency component. The two channels are followed by the first hidden layer, with  $N_{11}$  and  $N_{12}$  neurons connected to two inputs, respectively. Afterwards, there are  $H$  hidden layers and an output layer. The MC-DNN isolates the two frequency components, and better learns the respective classification features for each component.

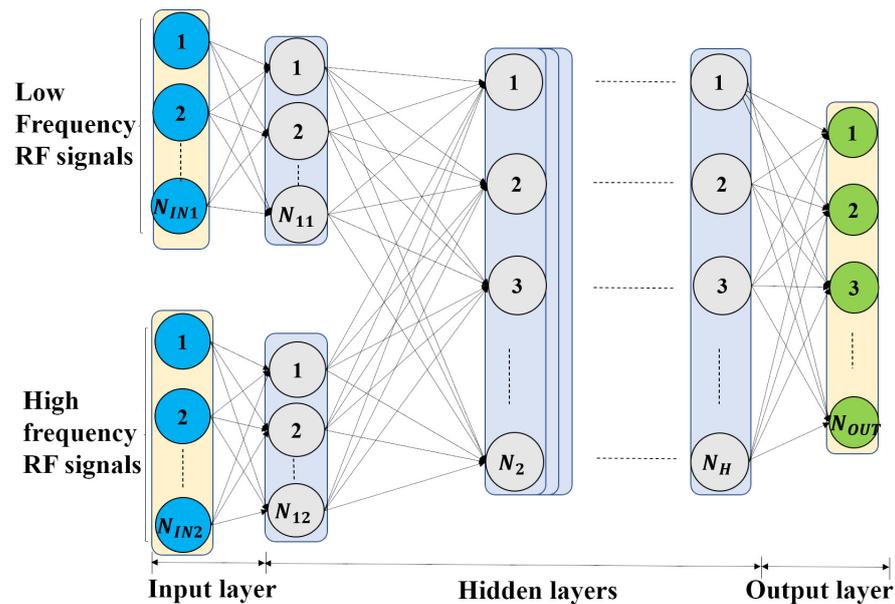


Figure 3. The structure of multi-channel deep neural network (MC-DNN).

#### 4.2.3. Learning Rate Decay

Learning rate is a significant hyperparameter in training a DNN since it defines the step size the DNN parameters update every time. If the learning rate is set too large, the parameters can learn too fast and oscillate around the optimal loss function minimization point without converging. On the contrary, if too small, the parameters can learn too slowly and overfit the training data. Both situations affect the DNN classification performance. The best choice of defining the learning rate is to set it large at first and reduce it gradually. This means the DNN learns fast at first and slowly when approaching the optimal minimization point. Thus, the learning rate cosine decay technique [46] is adopted, where the learning rate decreases as follows.

$$\eta_t = \frac{1}{2} \left( 1 + \cos\left(\frac{t\pi}{T}\right) \right) \eta, \quad (4)$$

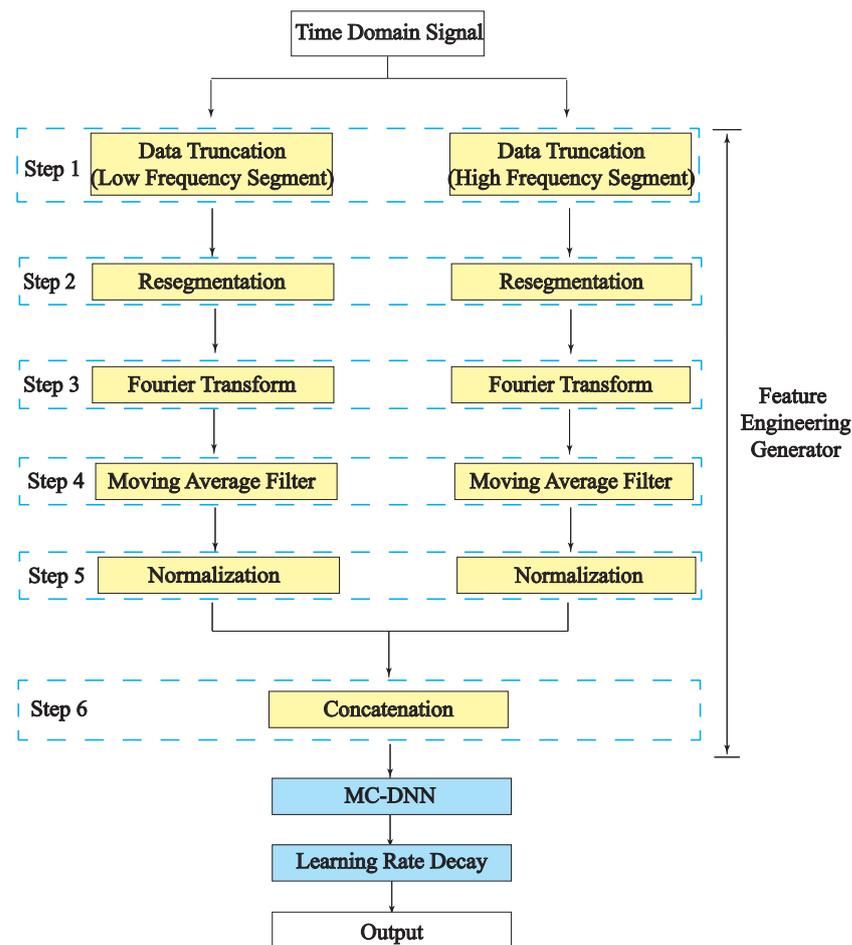
where the total number of epochs is  $T$ ,  $\eta$  is the initial learning rate, and  $\eta_t$  is the learning rate at epoch  $t$ . The learning rate decreases from an initial value  $\eta$  to approximately 0 following the cosine function. The speed of cosine decay is slow at the beginning, linear in the middle, and slow again at the end. This training technique enables the MC-DNN to learn fast at first and converge to the loss function minimization point in the end.

## 5. Experiments

In this section, the dataset in [22] is used to verify the effectiveness of our method. First, the details of the dataset are introduced. Then, our method is applied step by step, and the performance is compared. The final result of our method is also contrasted with other methods.

We use the performance of DNN and data without preprocessing as the baseline. Every other step of FEG preprocessing is applied based on previous ones. For example, the second preprocessing technique introduced is the data truncation and normalization, so the truncated and normalized data input into DNN is the second experiment. After cumulatively applying the FEG steps, the preprocessed data is input into MC-DNN for experiments.

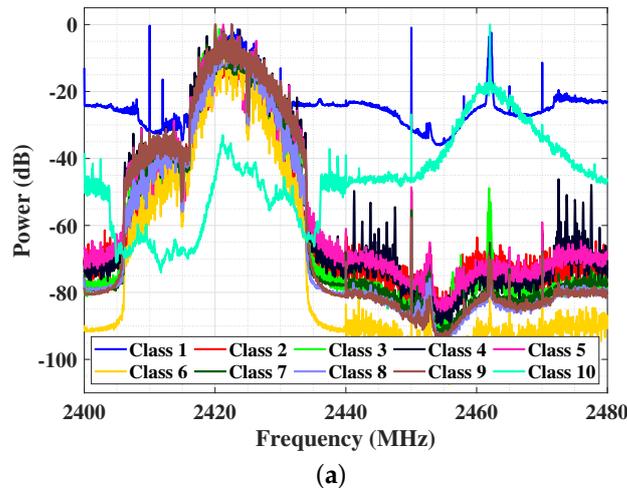
The entire system model of the FEG and MC-DNN is presented in the flow chart in Figure 4. The FEG preprocessing steps in the flow chart implemented for each experiment are carefully labeled.



**Figure 4.** The overall flow chart of the proposed method including FEG and DNNs for classification. The RF signals are preprocessed by the steps of FEG, and then input into DNNs for classification.

### 5.1. Dataset

To verify the effectiveness of our method, the dataset for UAV detection in [22] is adopted. It consists of data from three different types of UAVs: Parrot Bebop, Parrot AR Drone, and DJI Phantom 3. Each type of UAV has four flight modes: mode “On”, mode “Hovering”, mode “Flying without video recording”, and mode “Flying with video recording”. The dataset contains 10 categories of RF signals: background with no UAVs, four flight modes of UAV “Parrot Bebop”, four flight modes of UAV “Parrot AR Drone” and mode on of UAV “DJI Phantom 3”. Each category of data is collected by two RF receivers that intercept the UAV’s communications in each flight mode simultaneously. Because the RF receivers have bandwidth constraints, two receivers record low-frequency and high-frequency components, respectively. Then, the received time-domain data is labeled and stored subsequently. The originally captured data segments have  $10^7$  samples per segment. Next, the segments are divided into smaller segments with  $10^5$  samples to increase the amount of data for further supervised learning. The segments with  $10^5$  samples are processed by Fourier Transform into the frequency domain. The high-frequency component and low-frequency component are then concatenated. To ensure the concatenation to be continuous, the first 10 samples of the high-frequency component are multiplied by a coefficient determined by the low-frequency component. The final dataset spectra is shown in Figure 5a.



		42.0%	99.8%	21.5%	28.8%	25.9%	29.7%	26.2%	30.7%	45.1%	67.7%	44.8%
		58.0%	0.2%	78.5%	71.2%	74.1%	70.3%	73.8%	69.3%	54.9%	32.3%	55.2%
1	99.8%	4094	4	0	1	0	0	0	0	0	0	3
	0.2%	18.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.2%
2	21.5%	1	310	24	4	2	16	195	206	31	2	39.2%
	78.5%	0.0%	1.4%	0.1%	0.0%	0.0%	0.1%	0.9%	0.9%	0.1%	0.0%	60.8%
3	28.8%	0	525	851	693	591	534	1	6	184	434	22.3%
	71.2%	0.0%	2.3%	3.7%	3.1%	2.6%	2.4%	0.0%	0.0%	0.8%	1.9%	77.7%
4	25.9%	1	671	668	865	604	718	194	197	241	420	18.9%
	74.1%	0.0%	3.0%	2.9%	3.8%	2.7%	3.2%	0.9%	0.9%	1.1%	1.9%	81.1%
5	29.7%	0	76	135	103	499	132	20	28	9	263	39.4%
	70.3%	0.0%	0.3%	0.6%	0.5%	2.2%	0.6%	0.1%	0.1%	0.0%	1.2%	60.6%
6	26.2%	0	374	228	231	211	629	365	334	186	146	23.3%
	73.8%	0.0%	1.6%	1.0%	1.0%	0.9%	2.8%	1.6%	1.5%	0.8%	0.6%	76.7%
7	30.7%	1	7	3	4	1	1	466	386	67	2	49.7%
	69.3%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	2.1%	1.7%	0.3%	0.0%	50.3%
8	45.1%	1	12	3	10	5	4	761	864	63	7	49.9%
	54.9%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	3.4%	3.8%	0.3%	0.0%	50.1%
9	67.7%	0	12	12	14	6	27	75	39	1017	2	84.5%
	32.3%	0.0%	0.1%	0.1%	0.1%	0.0%	0.1%	0.3%	0.2%	4.5%	0.0%	15.5%
10	44.8%	2	109	176	175	181	39	23	40	2	821	52.4%
	55.2%	0.0%	0.5%	0.8%	0.8%	0.8%	0.2%	0.1%	0.2%	0.0%	3.6%	47.6%
		99.9%	14.8%	40.5%	41.2%	23.8%	30.0%	22.2%	41.1%	56.5%	39.1%	45.9%
		0.1%	85.2%	59.5%	58.8%	76.2%	70.0%	77.8%	58.9%	43.5%	60.9%	54.1%
			1	2	3	4	5	6	7	8	9	10

Figure 5. (a) The average spectra. (b) The confusion matrix. The spectra of the original RF signal dataset, which contains low-frequency component and high frequency component, and the confusion matrix of training DNN using the dataset. Class 1 is the background noise and class 2–10 are different flight modes of three kinds of UAVs.

5.2. DNN

The parameters for the DNN in Figure 5b are as follows. DNN has five layers in total, with one input layer, three hidden layers, and one output layer. Each hidden layer has 128 neurons and the output layer has 10 neurons, the same as the number of classes. The activation function for the output layer is the Softmax function, and the one for other layers is the ReLU function. The DNN is trained by an Adam optimizer, minimizing the mean square error loss function. The number of epochs is set to 300 and the batch size is 32. Note that the batch size is set to the powers of 2 to make calculations more efficient.

Using the dataset without signal FEG preprocessing as input to DNN, we get a baseline accuracy of 45.9% and an F1 score of 42.0%. The confusion matrix of the evaluation on the performance is shown in Figure 5b.

The details of confusion matrix plots are illustrated as follows (see Figure 5b for an example). Ten inner rows represent the output classes and ten inner columns represent the ten objective classes. The diagonal cells in green show the correct predicted samples and rate. Other cells in the inner rows and columns in red correspond to the number and portion of wrongly predicted samples. The top row and the leftmost column in yellow color

demonstrate the F1 scores of ten class predictions in green font and the complementary of F1 score in red font. The top leftmost cell in orange averages all the F1 scores and the complementary ones. Besides, the purple bottom row illustrates the recall in green font and FNR in red font. The purple rightmost column presents the precision in green font and the FDR in red font. The bottom rightmost cell in white reveals the average accuracy in black and the complementary error rate in red. The precision, recall, and F1 score can be calculated as follows.

$$\text{precision} = \frac{TP}{TP + FP} \quad (5)$$

$$\text{recall} = \frac{TP}{TP + FN} \quad (6)$$

$$\text{F1 score} = 2 \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{2TP}{2TP + FP + FN} \quad (7)$$

where TP means true positive, FP means false positive, and FN means false negative.

### 5.3. Joint DNN and Feature Engineering Generator

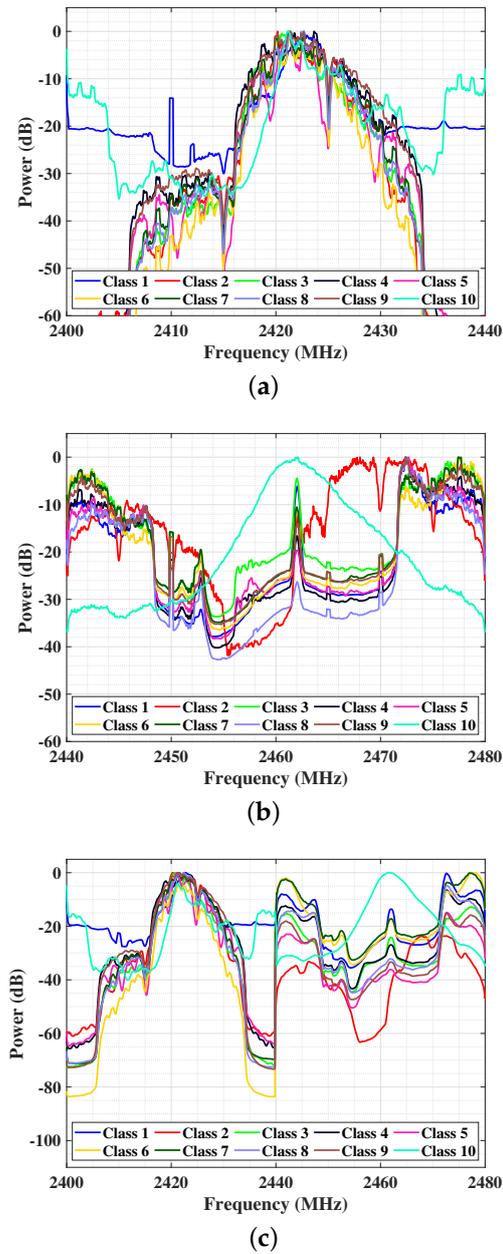
We evaluate each FEG preprocessing technique step by step cumulatively, and the data is input to the DNN for training.

#### 5.3.1. Data Truncation and Normalization

After applying data truncation and normalization, FEG preprocessing steps 1, 2, 3, and 5 in Figure 4 are implemented. The DNN is trained using the low-frequency component and high-frequency component separately, aiming at exploring more features contained in each component.

The low-frequency component of the data, plotted in Figure 6a, is first truncated and normalized. Note that the plotted low-frequency spectra are processed by a 10-point average filter for visualization. As in the figure, the region of signals between approximately 2415 MHz and 2435 MHz have similar trends and peaks, which makes the classification difficult. Only the two regions near 2400 MHz and 2440 MHz have some visible differences. The low-frequency component of data is used to train the DNN, which achieves an accuracy of 52.5% and an F1 score of 47.1%. The confusion matrix of training DNN using the normalized low-frequency component is plotted in Figure 7a. The accuracies of some categories of flight modes are around 25%, which means the category is not classified correctly. The performance requires further improvements.

The high-frequency component in Figure 6b is preprocessed in the same way as the low-frequency one, i.e., truncated and normalized. Note that the high-frequency component spectra plot is also processed by a 10-point average filter. Improved accuracy of 85.4% and F1 score of 84.1% are achieved by training DNN using the high-frequency component. The great improvement is because the high-frequency component has fewer similarities and more distinct features between the ten categories of signals. Besides, the data truncation and normalization avoid the high-frequency component from being dominated by the low-frequency component. The confusion matrix of using the high-frequency component to train is plotted in Figure 7b.



**Figure 6.** (a) Average spectra of the low-frequency component. Preprocessing steps 1, 2, 3, and 5 in the flow chart are implemented. (b) Average spectra of the high-frequency component. Preprocessing steps 1, 2, 3, and 5 in the flow chart are implemented. (c) Average spectra of the concatenation. Preprocessing steps 1–6 in the flow chart are implemented. The spectra of the low-frequency component, high-frequency component, and direct concatenation of 20-point moving average filtered low-frequency component and the 40-point moving average filtered high-frequency component. Class 1 is background noise and class 2–10 are different flight modes of three kinds of UAVs.



**Table 1.** The accuracies and F1 scores of DNN with low-frequency data and high-frequency data processed by  $n_l$  and  $n_h$  point moving average filters as inputs, respectively. Preprocessing steps 1–5 in the flow chart are implemented.

Data		n-Point Moving Average Filter										
		0	5	10	15	20	25	30	35	40	45	50
Low-frequency component	accuracy (%)	52.5	58.7	58.0	61.1	<b>65.5</b>	64.1	60.8	64.1	61.7	63.3	62.9
	F1 score (%)	47.1	53.1	52.0	56.7	<b>62.2</b>	59.5	55.8	60.5	56.8	59.7	59.2
High-frequency component	accuracy (%)	85.4	84.2	82.7	89.0	87.9	87.2	89.8	89.6	<b>90.6</b>	89.5	90.4
	F1 score (%)	84.1	82.8	81.2	88.0	86.8	85.8	88.8	88.5	<b>89.7</b>	88.4	89.4

### 5.3.3. Concatenation

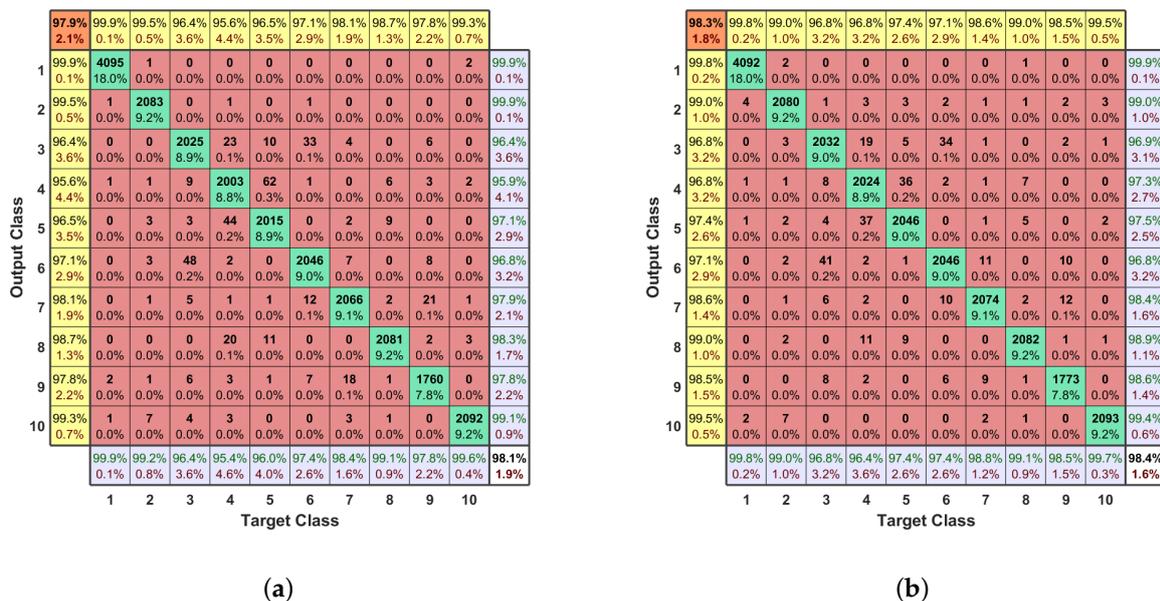
This subsection evaluates the effectiveness of concatenation (step 6) based on steps 1–5. All preprocessing steps 1–6 in FEG preprocessing are implemented. The low-frequency component filtered by the 20-point moving average filter and the high-frequency component filtered by the 40-point moving average filter are concatenated directly. The concatenated data is shown in Figure 6c, which has less information loss compared with the concatenation method in [22]. The accuracy of DNN trained with the concatenated data is 97.3% and the F1 score is 97.1%. The resulting confusion matrix is presented in Figure 7c. This proves that complete data details achieve better performance.

### 5.4. Joint MC-DNN and Feature Engineering Generator

The DNN is developed into the MC-DNN, and then the learning rate decay is added. The input in this section is the data fully processed by FEG.

#### 5.4.1. Multi-Channel Input

The designed MC-DNN in Figure 3 has double-channel inputs, the first hidden layer consisting of two parts for two inputs, three hidden layers, and one output layer. There are 256 neurons in the first hidden layer, and 128 neurons in other hidden layers. The result confusion matrix of using the preprocessed data to train MC-DNN is shown in Figure 8a, where the accuracy is improved to 98.1% and the F1 score is improved to 97.9%. This is because multi-channel input separates the inputs and makes the follow-up dense layers learn differently. Furthermore, two separate channels of inputs add more parameters in the MC-DNN, i.e., weights and biases. More parameters can better fit the complex relationships.



**Figure 8.** (a) The confusion matrix of the MC-DNN. (b) The confusion matrix of the MC-DNN with learning rate cosine decay. The confusion matrices of training MC-DNN and MC-DNN with learning rate decay using the fully FEG preprocessed data. Class 1 is background noise and class 2–10 are different flight modes of the three kinds of UAVs.

### 5.4.2. Learning Rate Decay

Cosine learning rate decay allows the MC-DNN to converge faster and fit the data. The initial learning rate  $\eta$  in Equation (4) is set to 0.01. After using the fully FEG preprocessed data as input, the result confusion matrix is presented in Figure 8b. The learning rate cosine decay training technique increases the accuracy to 98.4% and the F1 score to 98.3%.

### 5.5. Comparison

The performance comparison between FEG techniques and DNN structures is presented in Table 2. As shown in Table 2, the accuracy and F1 score are achieved with each preprocessing technique and DNN structure are additionally applied. Eventually, our method achieves an accuracy of 98.4% and an F1 score of 98.3% for the classification. Meanwhile, the FEG plays an important role in advancing the performance, improving from 45.9% to 97.3%. The DNN structure improve from 97.3% to 98.4%. The baseline accuracy and F1 score of training DNN are only 45.9% and 42.0% since techniques are not applied to extract and learn the signal features. The improved accuracy and F1 score verify the effectiveness of our method.

Our method also outperforms other methods in terms of accuracy and F1 scores [22–24] applying on the same dataset. The work in [22] offers an open dataset and designed a three-hidden-layer DNN for classification with the frequency-domain data as input. The proposed method classifies the presence and UAV types with accuracies of 99.7% and 84.5%. However, the overall ten-class accuracy and F1 score obtained are 46.8% and 43.0%. This accuracy may only prove the feasibility and not support accurate flight mode classification. Furthermore, Convolutional Neural Networks (CNN) are designed based on this dataset [23]. Dropout layers are added to prevent overfitting problem. Two separate CNN structures with different hyper-parameters are proposed for UAV detection and flight mode identification. The CNN structure for classification contains 6 one-dimensional (1D) convolutional layers and dropout rate of 0.2. The proposed model derives an accuracy of 59.2% and an F1 score of 55.1% for the ten-class classification. The multi-channel 1D CNN in [24] includes a feature extractor and a classical MLP. The captured 80 MHz

frequency spectrum is divided into 8 channels evenly, with each channel a separate input to the classifier. The multi-channel 1D CNN uses Adam optimizer and cross-entropy loss function. An accuracy of 87.4% and an F1 score of 77% are obtained by this model. The comparisons of accuracy and F1 score between our method and others are presented in Table 2.

Our method improves the accuracy and F1 score to 98.4% and 98.3%, respectively. Compared with other methods, our method first focuses on dataset preprocessing for full feature extraction. The preprocessing prepares the data for further MC-DNN learning. Moreover, the MC-DNN in our channel learns the low-frequency and high-frequency components separately, which contributes to the final performance. The additional learning rate decay is also effective for the convergence to good performance. Our method's high accuracy significantly reduces potentially high errors in classifying UAV modes. This leads to practically applicable solutions in real-world scenarios. From the comparison with other methods, the effectiveness and great performance improvement of the method in this paper are significant.

**Table 2.** The comparison of different combinations of FEG and DNN techniques and against existing methods. The FEG preprocessing techniques are implemented cumulatively, and the number represents the preprocessing steps in Figure 4. The operations corresponding to the steps are: step 1-data truncation, step 2-resegmentation, step 3-Fourier Transform, step 4-moving average filter, step 5-normalization, and step 6-concatenation. Since there are truncation and concatenation, the accuracies for low-frequency and high-frequency components are separately labeled in the brackets. The last three rows show the comparisons between our method and other methods.

Method	Accuracy		F1 score	
Unpreprocessed data + DNN		45.9%		42.0%
Preprocessing steps 1,2,3,5 + DNN	52.5% (low)	85.4% (high)	47.1% (low)	84.1% (high)
Preprocessing steps 1–5 + DNN	65.5% (low)	90.6% (high)	62.2% (low)	89.7% (high)
Preprocessing steps 1–6 + DNN		97.3%		97.1%
Preprocessing steps 1–6 + MC-DNN		98.1%		97.9%
Preprocessing steps 1–6 + MC-DNN + Learning rate decay		<b>98.4%</b>		<b>98.3%</b>
Classification method in [22]		46.8%		43.0%
Classification method in [23]		59.2%		55.1%
Classification method in [24]		87.4%		/

## 6. Conclusions

We proposed a joint approach of FEG and MC-DNN to detect UAV presence and classify UAV flight modes. The challenges of RF classification mainly focus on the high similarities between categories of RF UAV signals and the different characteristics represented by frequency components of data. To address these challenges, our method first preprocessed the RF signals by FEG using data truncation and normalization, moving average filter, and concatenation. A carefully designed MC-DNN with learning rate cosine decay, modified based on DNN, was proposed to classify the preprocessed data. The experiments showed the effectiveness of our method, which classifies ten categories with an accuracy of 98.4% and F1 score of 98.3%, and outperforms the state-of-the-art solutions. The proposed method could be extended by other researches on UAV detection and classification performance improvement, including more effective feature extraction as well as novel classification models focusing on finer frequency details.

**Author Contributions:** Conceptualization, S.Y. and Y.L.; methodology, S.Y., Y.L. and W.S.; investigation, S.Y. and C.G.; writing—original draft preparation, S.Y.; writing—review and editing, Y.L., W.M.,

W.S., and C.L.; supervision, Y.L. and C.L.; All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China grant number 61871096, and National Key R&D Program of China grant number 2018YFB2101300.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The dataset adopted is available online <https://data.mendeley.com/datasets/f4c2b4n755/1> (accessed in June 2020).

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

## Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
RF	Radio Frequency
FEG	Feature Engineering Generator
DNN	Deep Neural Network
MC-DNN	Multi-Channel Deep Neural Network
CNN	Convolutional Neural Network
NCA	neighborhood component analysis
RADAR	Radio detection and ranging
SNR	Signal-to-noise Ratio
SINR	Signal to Interference plus Noise Ratio
USRP	Universal Software Radio Peripheral
ReLU	rectified linear Unit Function
1D	One Dimensional
FDR	False Discovery Rate
FNR	False-Negative Rate

## References

- Zhao, N.; Lu, W.; Sheng, M.; Chen, Y.; Tang, J.; Yu, F.R.; Wong, K. UAV-Assisted Emergency Networks in Disasters. *IEEE Wirel. Commun.* **2019**, *26*, 45–51. [[CrossRef](#)]
- Ullah, S.; Kim, K.I.; Kim, K.H.; Imran, M.; Khan, P.; Tovar, E.; Ali, F. UAV-enabled healthcare architecture: Issues and challenges. *Future Gener. Comput. Syst.* **2019**, *97*, 425–432. [[CrossRef](#)]
- Singh, A.; Patil, D.; Omkar, S. Eye in the Sky: Real-Time Drone Surveillance System (DSS) for Violent Individuals Identification Using ScatterNet Hybrid Deep Learning Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018.
- Drummond, C.D.; Harley, M.D.; Turner, I.L.; AMatheen, A.N.; Glamore, W.C. UAV applications to coastal engineering. In Proceedings of the Australasian Coasts & Ports Conference 2015: 22nd Australasian Coastal and Ocean Engineering Conference and the 15th Australasian Port and Harbour Conference, Engineers Australia and IPENZ, Auckland, New Zealand, 15–18 September 2015; p. 267.
- Menouar, H.; Guvenc, I.; Akkaya, K.; Uluagac, A.S.; Kadri, A.; Tuncer, A. UAV-Enabled Intelligent Transportation Systems for the Smart City: Applications and Challenges. *IEEE Commun. Mag.* **2017**, *55*, 22–28. [[CrossRef](#)]
- Lygouras, E.; Gasteratos, A.; Tarchanidis, K.; Mitropoulos, A.C. ROLFER: A fully autonomous aerial rescue support system. *Microprocess. Microsyst.* **2018**, *61*, 32–42. [[CrossRef](#)]
- Idries, A.; Mohamed, N.; Jawhar, I.; Mohamed, F.; Al-Jaroodi, J. Challenges of developing UAV applications: A project management view. In Proceedings of the 2015 International Conference on Industrial Engineering and Operations Management (IEOM), Dubai, United Arab Emirates, 3–5 March 2015; pp. 1–10. [[CrossRef](#)]
- Zeng, Y.; Zhang, R.; Lim, T.J. Wireless communications with unmanned aerial vehicles: opportunities and challenges. *IEEE Commun. Mag.* **2016**, *54*, 36–42. [[CrossRef](#)]
- Zeng, Y.; Zhang, R. Energy-Efficient UAV Communication With Trajectory Optimization. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 3747–3760. [[CrossRef](#)]
- Luong, P.; Gagnon, F.; Tran, L.N.; Labeau, F. Deep Reinforcement Learning Based Resource Allocation in Cooperative UAV-Assisted Wireless Networks. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 7610–7625. [[CrossRef](#)]
- Li, B.; Zhao, S.; Zhang, R.; Yang, L. Full-Duplex UAV Relaying for Multiple User Pairs. *IEEE Internet Things J.* **2021**, *8*, 4657–4667. [[CrossRef](#)]

12. Horapong, K.; Chandrucka, D.; Montree, N.; Buaon, P. Design and use of “Drone” to support the radio navigation aids flight inspection. In Proceedings of the 2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC), St. Petersburg, FL, USA, 17–21 September 2017. pp. 1–6.
13. Sekander, S.; Tabassum, H.; Hossain, E. Multi-Tier Drone Architecture for 5 G/B5 G Cellular Networks: Challenges, Trends, and Prospects. *IEEE Commun. Mag.* **2018**, *56*, 96–103. [[CrossRef](#)]
14. Gupta, L.; Jain, R.; Vaszkun, G. Survey of Important Issues in UAV Communication Networks. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1123–1152. [[CrossRef](#)]
15. Nassi, B.; Shabtai, A.; Masuoka, R.; Elovici, Y. SoK - Security and Privacy in the Age of Drones: Threats, Challenges, Solution Mechanisms, and Scientific Gaps. *arXiv* **2019**, arXiv:1903.05155.
16. Zhi, Y.; Fu, Z.; Sun, X.; Yu, J. Security and privacy issues of UAV: A survey. *Mob. Networks Appl.* **2020**, *25*, 95–101. [[CrossRef](#)]
17. Güvenç, I.; Ozdemir, O.; Yapici, Y.; Mehrpouyan, H.; Matolak, D. Detection, localization, and tracking of unauthorized UAS and Jammers. In Proceedings of the 2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC), St. Petersburg, FL, USA, 17–21 September 2017; pp. 1–10. [[CrossRef](#)]
18. Ganti, S.R.; Kim, Y. Implementation of detection and tracking mechanism for small UAS. In Proceedings of the 2016 International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA, USA, 7–10 June 2016; pp. 1254–1260. [[CrossRef](#)]
19. Ezuma, M.; Erden, F.; Kumar Anjinappa, C.; Ozdemir, O.; Guvenc, I. Detection and Classification of UAVs Using RF Fingerprints in the Presence of Wi-Fi and Bluetooth Interference. *IEEE Open J. Commun. Soc.* **2020**, *1*, 60–76. [[CrossRef](#)]
20. Ezuma, M.; Erden, F.; Anjinappa, C.K.; Ozdemir, O.; Guvenc, I. Micro-UAV Detection and Classification from RF Fingerprints Using Machine Learning Techniques. In Proceedings of the 2019 IEEE Aerospace Conference, Big Sky, MT, USA, 2–9 March 2019. pp. 1–13.
21. Bhattacharjee, U.; Ozturk, E.; Ozdemir, O.; Guvenc, I.; Sichertiu, M.L.; Dai, H. Experimental Study of Outdoor UAV Localization and Tracking using Passive RF Sensing. *arXiv* **2021**, arXiv:2108.07857.
22. Al-Sa’ d, M.F.; Al-Ali, A.; Mohamed, A.; Khattab, T.; Erbad, A. RF-based drone detection and identification using deep learning approaches: An initiative towards a large open source drone database. *Future Gener. Comput. Syst.* **2019**, *100*, 86–97. [[CrossRef](#)]
23. Al-Emadi, S.; Al-Senaïd, F. Drone Detection Approach Based on Radio-Frequency Using Convolutional Neural Network. In Proceedings of the 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), Doha, Qatar, 2–5 February 2020; pp. 29–34. [[CrossRef](#)]
24. Allahham, M.S.; Khattab, T.; Mohamed, A. Deep Learning for RF-Based Drone Detection and Identification: A Multi-Channel 1-D Convolutional Neural Networks Approach. In Proceedings of the 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), Doha, Qatar, 2–5 February 2020; pp. 112–117. [[CrossRef](#)]
25. Moses, A.; Rutherford, M.J.; Valavanis, K.P. Radar-based detection and identification for miniature air vehicles. In Proceedings of the 2011 IEEE International Conference on Control Applications (CCA), Denver, CO, USA, 28–30 September 2011; pp. 933–940. [[CrossRef](#)]
26. Hong, T.; Fang, C.; Hao, H.; Sun, W. Identification Technology of UAV Based on Micro-Doppler Effect. In Proceedings of the 2021 International Wireless Communications and Mobile Computing (IWCMC), Harbin City, China, 28 June–2 July 2021; pp. 308–311. [[CrossRef](#)]
27. Jian, M.; Lu, Z.; Chen, V.C. Drone detection and tracking based on phase-interferometric Doppler radar. In Proceedings of the 2018 IEEE Radar Conference (RadarConf18), Oklahoma City, OK, USA, 23–27 April 2018; pp. 1146–1149.
28. Fu, H.; Abeywickrama, S.; Zhang, L.; Yuen, C. Low-Complexity Portable Passive Drone Surveillance via SDR-Based Signal Processing. *IEEE Commun. Mag.* **2018**, *56*, 112–118. [[CrossRef](#)]
29. Nijim, M.; Mantrawadi, N. Drone classification and identification system by phenome analysis using data mining techniques. In Proceedings of the 2016 IEEE Symposium on Technologies for Homeland Security (HST), Waltham, MA, USA, 10–11 May 2016; pp. 1–5.
30. Case, E.E.; Zelnio, A.M.; Rigling, B.D. Low-Cost Acoustic Array for Small UAV Detection and Tracking. In Proceedings of the 2008 IEEE National Aerospace and Electronics Conference, Dayton, OH, USA, 16–18 July 2008. pp. 110–113. [[CrossRef](#)]
31. Santos, N.P.; Lobo, V.; Bernardino, A. A ground-based vision system for UAV tracking. In Proceedings of the OCEANS 2015—Genova, Genova, Italy, 18–21 May 2015; pp. 1–9. [[CrossRef](#)]
32. Opromolla, R.; Fasano, G.; Accardo, D. A Vision-Based Approach to UAV Detection and Tracking in Cooperative Applications. *Sensors* **2018**, *18*, 3391. [[CrossRef](#)]
33. Behera, D.K.; Bazil Raj, A. Drone Detection and Classification using Deep Learning. In Proceedings of the 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 13–15 May 2020; pp. 1012–1016. [[CrossRef](#)]
34. Xu, C.; Chen, B.; Liu, Y.; He, F.; Song, H. RF Fingerprint Measurement For Detecting Multiple Amateur Drones Based on STFT and Feature Reduction. In Proceedings of the 2020 Integrated Communications Navigation and Surveillance Conference (ICNS), Herndon, VA, USA, 8–10 September 2020; pp. 4G1-1–4G1-7. [[CrossRef](#)]
35. Seltzer, M.L.; Yu, D.; Wang, Y. An investigation of deep neural networks for noise robust speech recognition. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 7398–7402.

36. Miao, Y.; Gowayyed, M.; Metze, F. EESSEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. In Proceedings of the 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Scottsdale, AZ, USA, 13–17 December 2015; pp. 167–174.
37. Toshev, A.; Szegedy, C. DeepPose: Human Pose Estimation via Deep Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 1653–1660.
38. Chan, T.; Jia, K.; Gao, S.; Lu, J.; Zeng, Z.; Ma, Y. PCANet: A Simple Deep Learning Baseline for Image Classification? *IEEE Trans. Image Process.* **2015**, *24*, 5017–5032. [[CrossRef](#)]
39. Bui, H.M.; Lech, M.; Cheng, E.; Neville, K.; Burnett, I.S. Object Recognition Using Deep Convolutional Features Transformed by a Recursive Network Structure. *IEEE Access* **2016**, *4*, 10059–10066. [[CrossRef](#)]
40. Medaiyese, O.; Ezuma, M.; Lauf, A.P.; Guvenc, I. Wavelet Transform Analytics for RF-Based UAV Detection and Identification System Using Machine Learning. *arXiv* **2021**, arXiv:2102.11894.
41. Ozturk, E.; Erden, F.; Guvenc, I. RF-Based Low-SNR Classification of UAVs Using Convolutional Neural Networks. *arXiv* **2020**, arXiv:2009.05519.
42. Zhang, H.; Cao, C.; Xu, L.; Gulliver, T.A. A UAV Detection Algorithm Based on an Artificial Neural Network. *IEEE Access* **2018**, *6*, 24720–24728. [[CrossRef](#)]
43. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaria, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* **2021**, *8*, 1–74. [[CrossRef](#)] [[PubMed](#)]
44. Rodriguez, J.D.; Perez, A.; Lozano, J.A. Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 569–575. [[CrossRef](#)] [[PubMed](#)]
45. Foody, G.M. Status of land cover classification accuracy assessment. *Remote. Sens. Environ.* **2002**, *80*, 185–201. doi: 10.1016/S0034-4257(01)00295-4. [[CrossRef](#)]
46. He, T.; Zhang, Z.; Zhang, H.; Zhang, Z.; Xie, J.; Li, M. Bag of Tricks for Image Classification with Convolutional Neural Networks. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 558–567. [[CrossRef](#)]