

RF-BASED LOW-SNR CLASSIFICATION OF UAVS USING CONVOLUTIONAL NEURAL NETWORKS

Ender Ozturk¹, Fatih Erden¹, Ismail Guvenc¹

¹Electrical and Computer Engineering, NC State University, Raleigh, NC 27606, United States

NOTE: Ender Ozturk, eozturk2@ncsu.edu

Abstract – Unmanned Aerial Vehicles (UAVs), or drones, which can be considered as a coverage extender for Internet of Everything (IoE), have drawn high attention recently. The proliferation of drones will raise privacy and security concerns in public. This paper investigates the problem of classification of drones from Radio Frequency (RF) fingerprints at the low Signal-to-Noise Ratio (SNR) regime. We use Convolutional Neural Networks (CNNs) trained with both RF time-series images and the spectrograms of 15 different off-the-shelf drone controller RF signals. When using time-series signal images, the CNN extracts features from the signal transient and envelope. As the SNR decreases, this approach fails dramatically because the information in the transient is lost in the noise, and the envelope is distorted heavily. In contrast to time-series representation of the RF signals, with spectrograms, it is possible to focus only on the desired frequency interval, i.e., 2.4 GHz ISM band, and filter out any other signal component outside of this band. These advantages provide a notable performance improvement over the time-series signals-based methods. To further increase the classification accuracy of the spectrogram-based CNN, we denoise the spectrogram images by truncating them to a limited spectral density interval. Creating a single model using spectrogram images of noisy signals and tuning the CNN model parameters, we achieve a classification accuracy varying from 92% to 100% for an SNR range from -10 dB to 30 dB, which significantly outperforms the existing approaches to our best knowledge.

Keywords – Convolutional neural networks (CNN), low SNR regime, RF fingerprinting, spectrogram, UAV classification

1. INTRODUCTION

Unmanned aerial vehicles (UAVs) or drones have recently gained a great deal of interest among researchers due to unrivaled commercial opportunities in various fields, such as wireless communications, logistics, delivery, search and rescue, smart agriculture, surveillance, among others [1]. In addition, the recent COVID-19 outbreak revealed the importance of remote operations in every aspect of life, which may accelerate social acceptance of drone use cases such as delivery of goods and medication [2, 3, 4]. With the new advances in airspace regulations and drone-related technologies, it is expected that there will be more and more UAVs in the skies for various use cases, sharing the airspace with other aerial vehicles [5]. The increase in daily drone usages can be considered in the context of The Internet of Everything (IoE), a broader term than Internet of Things (IoT), aiming to include the entire realm of information sources and destinations in one paradigm.

Innate advantages of UAVs that make them popular, such as ease of operation and low cost, could also be considered as major disadvantages from security and privacy perspectives. This motivates detection, classification, and tracking of different types of UAVs, and interdicts unauthorized or malicious UAVs to maintain privacy and security. Classification of UAVs can also be critical for forensics use cases, e.g. for identifying a UAV after a malicious activity (e.g. eavesdropping, espionage) based on the captured signals of the UAV. There have been many

criminal activities recently with drones involved, and their small sizes make it difficult to detect, classify, and interdict them [6, 7]. Latest surveys also demonstrate that 75% of the subjects exhibit privacy and security concerns about all unmanned aerial use cases [8]. In this regard, Federal Aviation Agency (FAA) of the United States recently announced a Proposed Rule that elaborates the future action that would require remote identification of unmanned aircraft systems to address safety and security concerns [9]. UAVs can be identified through a set of features that uniquely represent them. These features can be extracted from various data sources, such as visual data, acoustic, RF, or radar signals. Each of these source types has its own pros and cons which we will review in the next section. Our contributions with this work are summarized below.

- In this study, we develop a Convolutional Neural Network (CNN)-based classifier using both time-series signal images and spectrogram images of 15 different drone controller RF signals to classify drones of different makes and models. These signals are transmitted by proprietary circuit designs and contain distinct fingerprints of commercially available drones; they can be exploited by a machine learning model to classify the make and model of the drone. We use controller signals as the data set [27] was already in possession; however, the proposed approach can also be directly applied to the signals transmitted from drones to their controllers. A flowchart of the overall procedure is given in Fig. 1.

Table 1 – Related work on detection and classification of drones using ML techniques.

Literature	Source type	Features	Data process method	Classification	# of UAVs	Accuracy	Noise consideration
[10]	Drone RF signals	Slope, kurtosis skewness	Several ML algorithms	X	N/A	96.36%	X
[11]	Drone RF signal	CSI data	Channel state information	X	N/A	86.6%	X
[12]	Acoustic waves	MFCC and LPCC	SVM	X	N/A	96.7%	X
[13]	Acoustic waves	STFT features	CNN	X	N/A	99.87%	X
[14]	Camera images	RGB arrays	CNN for moving body detection and kNN for detection	X	N/A	93%	X
[15]	Camera images	RGB arrays	CNN on ZF and VGG16 and Fast R-CNN	X	N/A	0.66 mAP	X
[16]	Radar signals	Spectrogram	2-D complex-log-Fourier transform	X	N/A	3.27% EER	X
[17]	Radar signals	Range Doppler Matrix	SVM	X	N/A	98%	X
[18]	Radar signals	Micro-Doppler signature	PCA feature extraction on spectrograms	✓	3	94.7%	X
[19]	Radar signals	Micro-Doppler spectrogram	CNN and LSTM-RNN	✓	5	97.7%	X
[20]	Radar signals	Micro-Doppler signature	CNN	✓	6	94.7%	X
[21]	Radar signals	Micro-Doppler signatures through EMD	SVM	✓	11	>95%	X
[22]	Radar signals	Micro-Doppler signatures	SVM	✓	11	95.4%	X
[23]	Radar signals	Range Doppler spectrum	CNN	X	N/A	99.5% and 54.2% for 0 dB	✓
[24]	Drone RF signals	Statistical features e.g., mean, median, RMS	Logistic regression	✓	8	88-94% in 0.35 s	X
[25]	Radar signals	Micro-Doppler signature	ANN on MLP	✓	4	Various	✓
[26]	Controller RF signals	Shape factor, kurtosis, variance	Several ML algorithms	✓	17	98.13% and 40% for 0 dB SNR	✓
This work	Controller RF signals	Time-series signal and spectrogram RGB arrays	CNN	✓	15	99.7% and 99.5% for 0 dB SNR	✓

- For the classification tasks that involve RF fingerprinting, variations in the Signal-to-Noise Ratio (SNR) of the received RF signals is a challenging problem. In this work, we also address this practical problem by considering a range of SNR levels from -10 dB to 30 dB while training the CNN models. Noisy training data is generated by adding artificial white noise to the original data. When using spectrogram images to train the CNN models, we only focus on the frequency range of interest, which improves classification accuracy significantly in comparison with time-series images.
- In this work, we apply denoising on the spectrogram images to further improve the performance at low SNRs. We tune the spectral density level that will appear on the spectrogram image and filter out spectral densities lower than the tuned level. Our proposed classifier highly outperforms previously published work, especially at low SNRs.

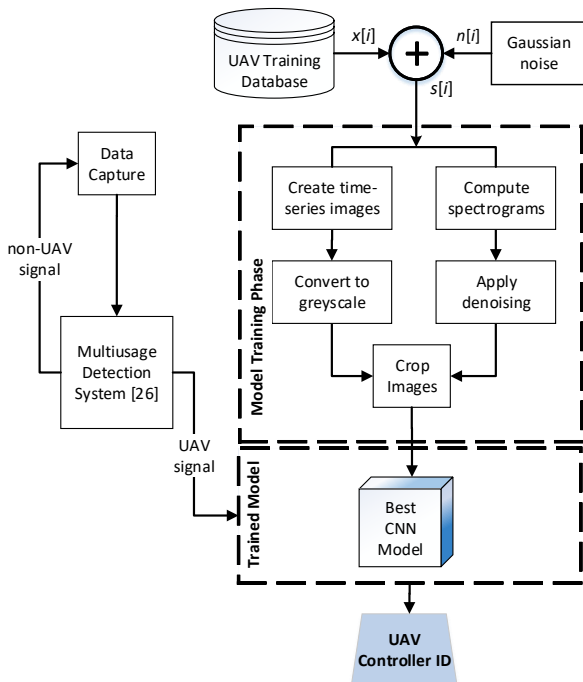


Fig. 1 – Overview of the proposed system. Multistage detector classifies the captured data as of type *UAV* or *non-UAV*. In the case of a *UAV* signal, captured data is artificially noised, and time-series and spectrogram images are created afterwards for training the corresponding CNN models. Time-series images are converted to grayscale to increase computational efficiency. Spectrograms are denoised to increase the model accuracy. Separate CNN models are trained and predictions are made using these CNN models. The best CNN model is deployed at the end.

A possible use case of the proposed system would be about the upcoming FAA regulation on Remote ID [9]. Remote ID is defined as the ability of a UAV to provide the relevant identity information to other parties. Even the drones will be obliged to reveal their IDs to comply with this regulation, it will still be possible for the malicious drones to fake their IDs. The system proposed in this work can be a part of a framework that verifies the drone IDs and make sure that the flying drone has the same ID as in the FAA's logs. This way countermeasures can be taken in the presence of a threat.

With regard to the type of images used, even though CNN models trained on spectrogram images perform better than models trained on time-series images for every scenario, we kept the results for the latter to provide a better basis for comparison of our contribution. This is because our present work is an extension of the work in [26], where statistical features extracted from time-series data have been used previously.

The rest of the paper is organized as follows. In Section 2, a comprehensive literature review including the information of noise consideration is given. In Section 3, the data set and the procedure for obtaining noisy samples are introduced.

Section 4 discusses an image data preprocessing step and the CNN-based classifier used in this work. Experimental results and relevant discussions are presented in Section 5. Finally, the paper is concluded in Section 6.

2. LITERATURE REVIEW AND CONTRIBUTIONS

Various approaches have been proposed in the literature for the detection and classification of drones. In Table 1, we summarize the related literature on drone detection and classification with some representative work and emphasis on the number of UAVs considered, classification accuracy, and noise considerations. Here we use the term *detection* as a special case of classification that has only two classes (i.e., *UAV/non-UAV*). Techniques used to achieve these tasks can be categorized based on the type of data being captured (e.g., radar signals, drone or controller Radio Frequency (RF) signals, acoustic data, or camera images), features extracted from the data (e.g., RF fingerprints, spectrogram images), and the Machine Learning (ML) algorithms deployed for classification. Acoustic sensors do not require line-of-sight (LOS); however, they suffer from short range, as drones could operate very quietly [12, 28], and data gathered using microphone systems are prone to wind and environmental clutter. On the other hand, a LOS vision under daylight is essential for techniques that utilize camera images [14, 29]. Using thermal or laser-based cameras to overcome this issue increases the cost significantly.

Radar signals are immune to environmental factors, such as acoustic noise and fog. However, drones are small devices with tiny propellers which make it hard to perceive and distinguish them from each other by most radars. A high-frequency wideband radar could be used to deal with these difficulties [20, 30, 31, 18]. Such radars are considerably expensive and suffer from high path loss. RF signals of either drones themselves or controllers are mostly at sub-6 GHz band and share unlicensed Wi-Fi bands. As a result of this, equipment to capture RF signals are affordable, but on the downside, RF-based techniques require special attention for handling interference from other co-channel signal sources. Besides, no LOS is required, and these techniques are immune to many problems that acoustic and visual techniques suffer from.

RF signals can be used for classification of the UAVs, either directly or indirectly after some processing. In [26, 10, 24], time-domain statistical properties of the RF signal, such as slope, kurtosis, skewness, shape factor and variance, are used as features along with different ML algorithms to detect and classify drones. However, since unlicensed bands are heavily employed, time-domain information suffers from low SNR. Frequency-domain representation of RF signals can also be used to distinguish

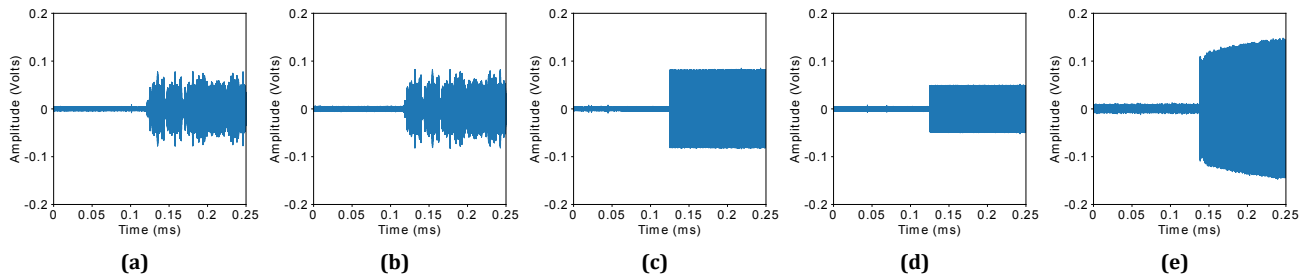


Fig. 2 – Sample controller time-series RF signals: (a) DJI Matrice 100, (b) DJI Matrice 600, (c) Spektrum DX5e, (d) FlySky FS-T6, and (e) Spektrum JR X9303. RF signals from different controllers may look alike, making it difficult to identify the drones based on only the envelopes of the captured signals.

between different types of drones. Transforming RF signals into the frequency domain filters out the out-of-band noise and helps improve classification accuracy up to a certain extent.

In the literature, there are studies using radar signals and spectrograms to detect and classify drones [16, 18, 19, 20, 21, 22, 23]. However, there is no study that utilize spectrograms of RF signals in the context of UAV detection/classification to the best of our knowledge.

Even though the mass majority of classification efforts in this field aim to identify drone make and model to support a decision of friend/foe, there are some other work that use ML techniques to identify drone pilots. For example, in [32], drone controller RF signals are recorded to characterize pilot activity, and different types of maneuvers that a pilot could do are used as features.

Classification accuracy should be considered together with the number of UAVs as it gets harder to classify UAVs with high accuracy as the number of classes increases. For studies which have *X* marks in the *Classification* column, the proposed models performed only *detection* which means there are only two classes. We also provide the information about whether the work considers noise or not, to better emphasize our contribution.

3. DATA SET AND NOISING PROCEDURE

In this work, the data set in [26] is used. This data set consists of RF signals from 15 different off-the-shelf UAV controllers listed in Table 2. RF signals were captured using an Ultra-Wideband (UWB) antenna and an oscilloscope with a sampling rate of 20 Gsa/s. Total number of samples in each signal is 5×10^6 , which corresponds to a time duration of 250 μ s. Time-series and spectrogram images are created from the training RF signals, and CNN models are generated for each image database.

3.1 Image creation process

The time-series RF signal of a controller is kept in a 1-D array. Time-series images are simply acquired by plotting these 1-D arrays. RF signals captured from different UAV controllers are illustrated in Fig. 2. As it can be

Table 2 – UAV controllers used in this work.

UAV ID (#)	Brand & Model
1	Jeti Duplex DC-16
2	DJI Matrice 100
3	DJI Matrice 600
4	DJI Phantom 3
5	DJI Inspire 1 Pro
6	Spektrum DX5e
7	Spektrum DX6e
8	FlySky FS-T6
9	Futaba T8FG
10	Graupner MC-32
11	Hobby King HK-T6A
12	Spektrum JR X9303
13	DJI Phantom 4 Pro
14	Spektrum DX6i
15	Turnigy 9X

observed from the figure, RF signals exhibit different waveforms. Digital image processing literature bestows useful techniques to distinguish such signals using an envelope detector and template matching-based approaches [33]. However, some controller signals may exhibit similar envelopes (e.g., RF signals in Fig. 2(a) and Fig. 2(b), or the signals in Fig. 2(c) and Fig. 2(d)), making it challenging to identify the controllers with these approaches. Besides, taking into account that signal envelopes get distorted at high noise levels, more advanced approaches are needed to achieve high classification accuracy.

Spectrogram images are created calculating power spectral densities of the signals using Welch’s average periodogram method, which is also called Weighted Overlapped Segment Averaging (WOSA) method [34]. In this method, time-domain signal $x[i]$ captured from a UAV is divided into successive blocks and averaged to estimate the power spectral density after forming the periodograms for each block, i.e.,

$$x_m[i] = w[i]x[i + mR], \tag{1}$$

where $i = 0, 1, \dots, M - 1$ is the sample index, M is the window size, $m = 0, 1, \dots, K - 1$ denotes the window index, K is the total number of blocks, R is the window’s

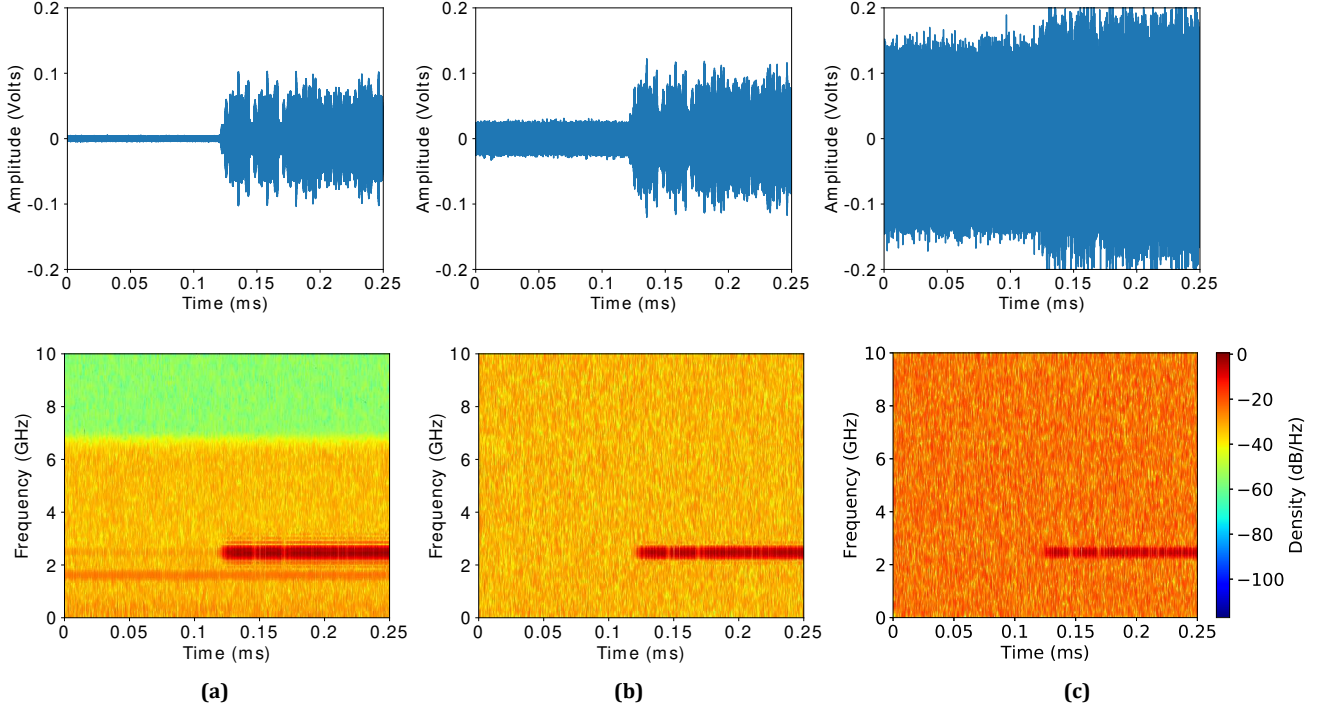


Fig. 3 – Noisy controller signal of a DJI Inspire 1 Pro for different SNRs: a) 30 dB, b) 15 dB, and c) 0 dB. Base noise in lab environment is around 30 dB. As SNR decreases, distortion occurs for both image types. All spectrogram images have the same density color scale.

hop size that tunes the amount of overlap between consecutive windows, and $w[i]$ is the window function. Then the periodogram of a block is calculated as

$$\begin{aligned} P_{x_m, M}(w_k) &= \frac{1}{M} |FFT_{N, k}(x_m)|^2 \\ &= \frac{1}{M} \left| \sum_{i=0}^{N-1} x_m[i] e^{-2j\pi ik/N} \right|. \end{aligned} \quad (2)$$

Consequently, Welch estimate of power spectral density is calculated as follows

$$\hat{S}_x^W = \frac{1}{K} \sum_{m=0}^{K-1} P_{x_m, M}(w_k). \quad (3)$$

In this paper the Hanning window is used while calculating periodograms. Then the calculated densities are mapped to a color scale to create spectrograms. We use a color map that spans the whole color space evenly, i.e., passes through all the colors in the visible range, which increases the accuracy of the proposed model significantly.

3.2 Noising procedure

Assuming fixed environmental noise, the SNR level of an RF signal decreases as the source gets farther away from the receiving antenna. Since the drone or the controller position and hence their distance to the receiver antenna may vary in different scenarios, systems that can work under low SNR regimes are required. In this study, we propose a method that can identify drones even at very

low SNRs. Since the data [26] is collected in a lab environment, the noise is stable and the same for all measurements. To train and test our models for noisy signals, we add white Gaussian noise to the raw data, and then generate the corresponding time-series and spectrogram images. While generating the noisy signals, we first use Higuchi's fractal dimension method [35] to find the approximate position of the transient signal segment. We use this information to distinguish between the noise and the RF signal and calculate their actual power separately as

$$P_{\text{noise}} [\text{dB}] = 10 \times \log_{10} \left(\frac{\sum_{i=0}^{T_b} |x[i]|^2}{T_b} \right), \quad (4)$$

and

$$P_{\text{signal}} [\text{dB}] = 10 \times \log_{10} \left(\frac{\sum_{i=T_e}^N |x[i]|^2}{N - T_e} \right), \quad (5)$$

where T_b and T_e are the indexes where the transient begins and ends, respectively.

Next, we calculate the SNR (Γ) of the received signal and the difference between the current SNR level and the desired SNR level as follows:

$$\Gamma_{\text{signal}} [\text{dB}] = P_{\text{signal}} - P_{\text{noise}}, \quad (6)$$

and

$$\Delta\Gamma [\text{dB}] = \Gamma_{\text{signal}} - \Gamma_{\text{desired}}. \quad (7)$$

Finally, an appropriate amount of random noise $n[i]$ is added to the whole signal to set the signal to the desired

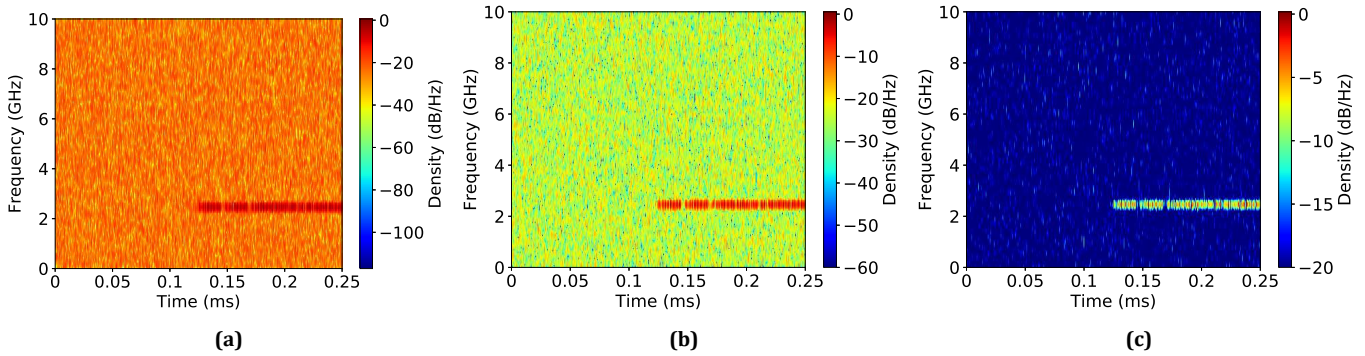


Fig. 4 – Denoising of a 0 dB SNR signal at different cut-off values: a) no truncation, b) -60 dB/Hz, and c) -20 dB/Hz. Truncation process sets the lower limit of the density color scales. This increases the level of representation of the high density signal components on the spectrogram image, consequently model accuracy is improved.

SNR level as

$$s[i] = x[i] + n[i], \tag{8}$$

where $n[i] = \Delta\Gamma \times \mathcal{N}(0, 1), i = 0, 1, \dots, N - 1$, and N is the total number of samples in $x[i]$. Note that $\Delta\Gamma$ that is used to generate the noise sequence is not in dB scale.

A set of artificially noised time-series images and the corresponding spectrograms are given in Fig. 3. Subject to the type of controller, the original data has an SNR of about 30 dB. Increased noise causes distortion visible in both image types. However, time-series images are affected more. Spectrograms preserve signal characteristics better than time-series images as signal components can be better resolved in the frequency domain.

4. IMAGE PREPROCESSING AND CNN-BASED UAV CLASSIFICATION

CNN is a deep learning algorithm which has been proven to perform well in image recognition and classification tasks [36]. CNNs have layers just as any other neural networks; however, different from other deep learning algorithms, convolution layers are used to apply various filters to an image to extract features no matter at which part of the image they reside. This nature of the algorithm makes CNN a perfect fit for 2D data (e.g., images), and also reduces the number of required weights in a neuron, thus yields lower computational complexity in comparison with conventional deep neural network architectures. In this work, spectrograms and time-series images of RF signals have been used as inputs to the CNN models.

Even though CNN is a very powerful approach for extracting features from images, the preprocessing phase of the source data is crucial to increase the overall success of the classification and decrease the computational cost.

4.1 Conversion to grayscale and image cropping

As reviewed in Section 3.1, spectrograms reflect the power spectral densities of the signals. Since color depth preserves distinctive information, these images should be

kept in red/green/blue (RGB) format. However, time-series images are not represented in such a format, and therefore, to decrease the complexity, time-series signal images should be converted to grayscale if these images do not come in grayscale by default.

Time-series and spectrogram images typically have axes, ticks and labels regardless of the software tool that is used to create them. We remove all those parts before beginning post-processing the images. Besides, captured images would include both the noise-only signal (when there is no transmission) and the transmitted signal (see the time-series signals in Fig. 3). By using Higuchi’s fractal dimension method as suggested in Section 3.2, it is possible to remove out the noise-only part in both image types. In addition, one of the axes of the spectrograms will includes frequency domain information. In case the frequency range of interest is known, it is appropriate to crop the spectrograms further to lower the computational cost focusing on the desired frequency band only.

4.2 Denoising the spectrograms

Denoising is an important step towards improving the accuracy of the spectrogram image-based classification. Denoising by truncation is only applied to spectrogram images. Power spectral densities should be calculated up to a certain frequency that is defined by the sampling rate for several instants in the time domain that covers the whole signal. These spectral density values are mapped to the RGB color scale while creating spectrograms: the minimum and the maximum spectral densities are mapped to the coolest and warmest colors of a chosen color map, whereas the colors for the intermediate values are adjusted accordingly. In order to denoise the spectrogram, a cut-off density is picked as a threshold, and the spectrogram is truncated by setting the elements of the spectral density array that are smaller than this cut-off to the cut-off value itself. This process assures signal components with smaller densities to be cleared. Since most of the noise components have lower power densities than the drone controller signal itself for a wide range of SNRs,

the truncation process is essentially a denoising procedure. The rest of the signal is mapped to the same color range set, which increases the level of representation of the details. As a result, non-noise (i.e., RF) signal components come forward that help the CNN models learn better. The procedure described above could be explained mathematically as follows:

$$S'[f, i] \xrightarrow{f_t} \begin{cases} \gamma_c, & \text{if } S[f, i] \leq \gamma_c, \\ S[f, i], & \text{else} \end{cases}, \quad (9)$$

$$S'[f, i] \xrightarrow{f_c} r_{f,i}, g_{f,i}, b_{f,i},$$

where f_t is the truncation function, $S'[f, i]$ is the truncated signal subject to the cut-off value γ_c , f_c is the color mapping function, and $r_{f,i}$, $g_{f,i}$, and $b_{f,i}$ are the color intensities in the corresponding channels.

There exists a critical trade-off that depends on the chosen cut-off threshold. For a given SNR level of the signal in hand, spectrograms should be truncated at an optimum level for that SNR. More specifically, in the case of under-denoising, excess noise causes overfitting, while in the case of over-denoising, useful information is wiped out together with the noise, which yields to underfitting. To illustrate this trade-off, we will consider Fig. 4 which shows the spectrograms of a DJI Inspire 1 Pro controller signal, that is artificially noised to 0 dB SNR, at different truncation levels. In this figure, it is observed that as the threshold increases (i.e., from no truncation to -20 dB/Hz), the lower limit of the density on the spectrograms changes. This lowest density is the lowest value in the domain set. As a result of truncation, high density components of the signals are represented better on the images.

Another aspect of creating CNN models for different data sets at various SNR levels is the necessity of defining the SNR of a signal beforehand to invoke the appropriate model. To manage this, we propose to follow two approaches while working with the spectrograms. In the first approach, we create and optimize different CNN models for different SNR levels. The idea is that, assuming that the captured signal's SNR can be measured, the model having the closest SNR is called to perform classification. Even though, calculating the SNR of a received signal in real time is a tricky task, we believe, with the help of featured state-of-the-art measurement devices and newly developed algorithms [37], this would not be a problem. In the second approach, we define an optimum cut-off value (i.e., minimum average validation loss among all different cut-offs), merge all the images of different SNR levels truncated at this level to create a new comprehensive data set, and then train a single model. The major advantage of this second approach is that it is no longer required to determine the SNR of the signal in advance.

4.3 Training and testing CNN models

In this work, CNNs are trained using Keras with TensorFlow at the backend. In the models created, we have three convolution layers (Conv2D) followed by pooling layers (MaxPool2D) and then a fully connected layer followed by the output layer. Convolution layers get deeper (i.e., the number of filters increase), and size of the images get smaller as the data travels deep into the model, in accordance with the general convention. The CNN models have been trained and tested with 3 : 1 ratio for each UAV class. Optimum hyperparameters are determined after running a vast amount of simulations. Results are presented in the next section.

An illustration of the CNN architecture is shown in Fig. 5. While training the models, the categorical cross-entropy function is used as the loss function

$$\mathcal{L}(W) = -\frac{1}{N} \sum_{i=1}^N [y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})], \quad (10)$$

where W represents the model parameters, and $y^{(i)}$ and $\hat{y}^{(i)}$ represent the true labels and predicted labels for the i -th image, respectively. This function gets smaller as the true and the predicted results get closer to each other. The aim of the model is to find the optimum set of model parameters to minimize this function, i.e.,

$$\hat{W} = \underset{W}{\operatorname{argmin}} \mathcal{L}(W). \quad (11)$$

Probability of the i -th test image, expressed as $\mathbf{x}^{(i)}$ in vector form, being a member of the k -th class is calculated using normalized exponential function as:

$$p_k(\mathbf{x}^{(i)}) = \frac{e^{\hat{v}_k^{(i)}}}{\sum_j e^{\hat{v}_j^{(i)}}}, \quad (12)$$

where $\hat{\mathbf{v}}^{(i)}$ is the $K \times 1$ vector output of the final model that uses optimized weights given in ((11)), and K is the number of classes. The class that has the maximum probability is chosen to be the prediction of the model for the i -th test image, $\hat{y}^{(i)}$, for the given image

$$\hat{y}^{(i)} = \underset{k}{\operatorname{argmax}} p_k(\mathbf{x}^{(i)}). \quad (13)$$

The next section presents the experimental results that are acquired with the CNN models created for both the time-series and spectrogram images. Note that data sets used to train CNN models include either time-series images or spectrograms.

5. EXPERIMENTAL RESULTS

During training the CNN models, the original data set in [26], where the SNR is about 30 dB for the whole set, was used. We extended this data set by considering four

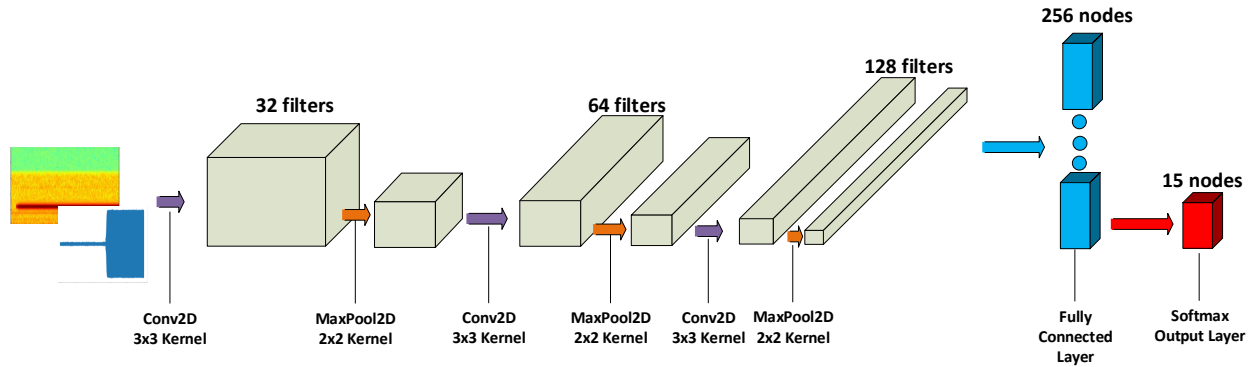


Fig. 5 – CNN architecture formed of three convolution and pooling layer pairs followed by a fully connected and a softmax output layer. The number of filters increase as data travels deep into the model to capture a widening variety of features better. Softmax output layer gives a set of predictions resolving the maximum likelihood of the signals class reference. The one with the highest probability is predicted by the model to be the class of the signal.

additional SNR levels ranging from 0 dB to 20 dB for time-series signal-based classification, and seven different additional SNR levels ranging from -10 dB to 20 dB for spectrogram-based classification, with SNR increments of 5 dB in both cases. To train the models, we created 100 images for each class and for each unique SNR truncation threshold pairs following the noising procedure described in Section 3.2 and the denoising procedure (for spectrogram images only) described in Section 4.2. Throughout the study, we created more than 100 data sets, each having 1500 images (15 classes with 100 images each). A Hanning window function of size 128 with 16 overlap samples is used while creating the spectrograms.

Before feeding the CNN, we crop the images appropriately to get rid of the unnecessary parts of the images and reduce the file sizes, which helps speed up the converging of the CNN models. Resulting spectrogram and time-series signal images have the sizes of $(90 \times 385 \times 3)$ and $(779 \times 769 \times 1)$, respectively. In this work, we used brute force searching to optimize CNN model parameters. We utilized the NC State University HPC (High Performance Computing) Facility to run parallel simulations for different sets of hyperparameters to find the optimum parameter set.

Note that, after rigorous simulations, the optimum activation function came up to be ReLu for all hyperparameter combinations. We also used single stride in both directions on images with no dilation, and valid padding for all models created regardless of SNR level and type of data. Remaining details of the models are given in Fig. 5, and Table 3 and Table 4.

In the rest of this section, we will first give considerations about the environmental interference issues and then present the classification results for the time-series images and spectrogram images. Subsequently, we will discuss the relation between classification accuracy and training set size and, finally, share the results for out-of-library classification performance of the proposed model.

Table 3 – Optimum set of hyperparameters for models trained on time-series images.

SNR (dB)	Optimizer	Batch size	Validation accuracy (%)
30	SGD	4	99.7
20	Adagrad	4	96.5
10	Nadam	16	81.6
5	Nadam	1	65.3
0	Adagrad	1	50.1

5.1 Comments on environmental interference

All the signals used in this study are recorded for a wide range of frequencies, i.e., 0–10 GHz, as illustrated by the spectrogram in Fig. 3(a). The first observation that can be made in there is that the frequency utilization significantly decreases above roughly 7 GHz, which is because there is no wireless transmission for that frequency range near the locations where we conducted the measurements. One can also notice the high color intensity at the GSM band around 1800 MHz. Since all of the drone controllers considered in this study transmit in the 2.4 GHz ISM band, notable densities in other bands on spectrograms have no effect on the model accuracy. However, the 2.4 GHz band is also used heavily by Wi-Fi and Bluetooth transmitters. In case Wi-Fi and/or Bluetooth signals are received, our proposed model applies a multistage detection system described in [26] to detect those type of signals and filter them out.

Raw data used in this work have been gathered in an indoor environment where Wi-Fi and Bluetooth signals could exist. A 24 dBi gain directional antenna has been used to capture the signals. It is known that IEEE 802.11 standards family routers implement Carrier-Sense Multiple

Table 4 – Optimum set of hyperparameters for models trained on spectrogram images.

SNR (dB)	Cut-off level (dB/Hz)	Optimizer	Batch size	Validation accuracy (%)
30	-100	Adamax	8	99.7
20	-90	Nadam	2	99.7
15	-10	Adam	32	100.0
10	-10	Nadam	2	100.0
5	-10	Adam	4	99.7
0	-10	Nadam	8	99.5
-5	-20	RMSProp	8	99.5
-10	-15	Nadam	16	92.0
Merged	-10	SGD	1	98.8
Merged*	-10	SGD	1	96.9

*Refers to the set of images created by assuming SNR levels different than the ones used to train the merged model.

Access (CSMA) techniques, which may help reducing the probability of interference with Wi-Fi transmitters when the drone controllers are close to the receiving antenna. Besides, low-power Bluetooth transmitters will not possess a high risk of severe distortion on the received signal. Moreover, our classifier makes a decision each time after processing a signal frame of $250 \mu\text{s}$. A short duration of signals allow our system to catch drone controller signals even in the existence of other packet-based communication technologies as they do not transmit packets continuously. While capturing a drone data-only signal frame may introduce time delays in identifying the drone, this delay will be on the order of milliseconds. Therefore, we can safely conclude that labeling the training set as if there are no WiFi and Bluetooth signals complies with real-world scenarios.

5.2 UAV classification using time-series images

We optimize five different CNNs for time-series images by a brute force searching approach. We ran simulations for each data set using all combinations of seven different optimizers, seven different batch sizes, and five different activation functions, which add up to 245 distinct simulations. The parameter set that gives the highest accuracy is chosen. Optimized parameters for these models are given in Table 3 for reproducibility. We observe that CNNs gather distinctive features from both the transient (i.e., the signal segment where the noise-only region ends and the RF signal begins) and the envelope of the RF signal. As the signal swamps into noise as SNR decreases, first the transient information disappears whereas the information carried in the signal envelope survives a little longer. When the SNR is further decreased, envelope information also disappears. Thus, the validation accuracy

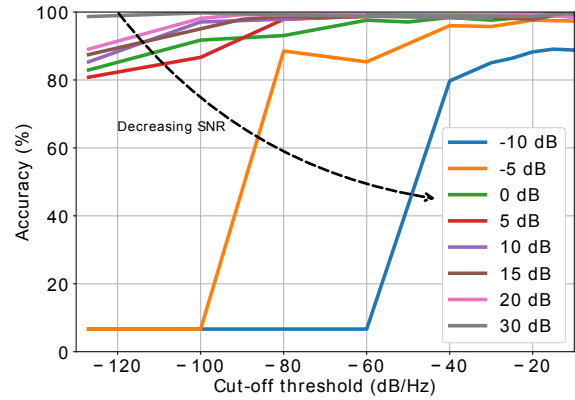


Fig. 6 – Spectrogram model classification accuracy versus the cut-off threshold for different SNR levels. Denoising the spectrograms by truncating the spectral densities, subject to a threshold, increases the model accuracy in general. Models trained with high-SNR data give reasonable accuracy even without denoising. Low-SNR models need to be denoised a priori.

drops from 99.7% to 50.1% as the SNR goes down from around 30 dB to 0 dB. Even though different optimizers could give the maximum accuracy for different SNR levels, all optimum models use Rectified Linear Unit (ReLU) as the activation function.

Both in-band and out-of-band noise cause distortion in time-series images of RF signals, and so the models trained on time-series images suffer from noise more than the models that use the spectrogram images. Besides, while using the time-series images, trained models extract features from the amplitude of the signals itself. However, the amplitude of a received signal depends on the distance between the receiver and transmitter antenna. This is an obvious problem when the only distinctive difference between the time-series signal images of any two controllers is the difference in their amplitude (e.g., see the RF signals in in Fig. 2(c) and Fig. 2(d)). It is worth noting that best results reported for the same number of classes in the previous work [26] that uses kNN (k-nearest neighbors), random forest and discriminant analysis techniques are slightly better for high SNR levels ($\approx 98\%$ vs 96.5% at 20 dB). However our results for models trained on time-series images are significantly better for low SNR levels ($\approx 40\%$ vs 50.1%). Moreover, models trained on spectrograms show even better performances. The next subsection is dedicated to results of models that employ spectrogram images.

5.3 UAV classification using spectrogram images

Two approaches are adopted while creating models on spectrogram images. In the first approach, we assumed that the SNR level of a received signal can be measured prior to classification and created different models for different SNR levels. In the second approach, we used a merged data set that includes spectrogram images of different SNRs to create a model that can be used to classify any received signal without any prior information about its SNR. Details of these approaches are given in the following subsections.

5.3.1 Models with single SNR training sets

We have created eight models for eight different SNR levels that are truncated at their own optimum levels. To use this approach, the SNR of the received signal should be calculated first, and then the model that has the closest SNR should be called to perform the classification. We observed that all of the models give their highest accuracy with the ReLu activation function. The sensitivity of the validation accuracy to a single output was found to be $0.27\% \text{ sample}^{-1}$. Optimized parameters of the models using spectrogram images are given in Table 4. It is seen that the lowest accuracy belongs to the SNR level -10 dB among the individual sets. Performances of all the other models can be considered almost perfect. It is also observed from Table 4 that the optimum cut-off levels are different for different SNR levels.

Classification accuracy at different truncation thresholds for different SNR levels are given in Fig. 6. By considering this figure and Table 4 together, one can conclude that, in general, the classification accuracy tends to increase with the increasing level of truncation. For high SNRs (i.e., 20 dB and 30 dB), spectral densities of the signals are much higher than the noise; therefore, truncating the images at different levels does not wipe out much information. As a result, the accuracy curve navigates flatter, and the necessary cut-off threshold is low (-100 dB/Hz and -90 dB/Hz). At medium SNRs (i.e., $0-15 \text{ dB}$), a high level of truncation is required to preserve as much information as possible (all -10 dB/Hz). On the other hand, at the lowest end of SNRs (i.e., -5 dB and -10 dB), without truncating the images, no learning occurs at all. For these lowest two SNRs, distinctive information in the spectrograms is swamped into noise so with no truncation, the accuracy is found to be only 6.66% . As the cut-off threshold increases, first a reasonable accuracy is acquired for a -5 dB SNR data set at the -80 dB/Hz threshold level. This amount of filtering is still not sufficient for -10 dB SNR, which only begins to learn at a comparably higher threshold of -40 dB/Hz . Moreover, the -10 dB/Hz threshold level gives lower accuracy than the models trained at medium SNRs (i.e., $0-15 \text{ dB}$) using the same threshold. This is because over-denoising chops the meaningful information together with the noise, and consequently, the optimum cut-off level is slightly lower than -10 dB/Hz (i.e., -20 dB/Hz for -5 dB SNR and -15 dB/Hz for -10 dB SNR). If the cut-off threshold is too high, this wipes out all the information, making all spectrograms look alike and consequently, there will be no learning.

The advantage of using spectral domain information could be seen from the results of a 0 dB SNR model where the classification accuracy for time-series images is only 50.1% (Table 3), whereas it is 82.9% (Fig. 6) for the spec-

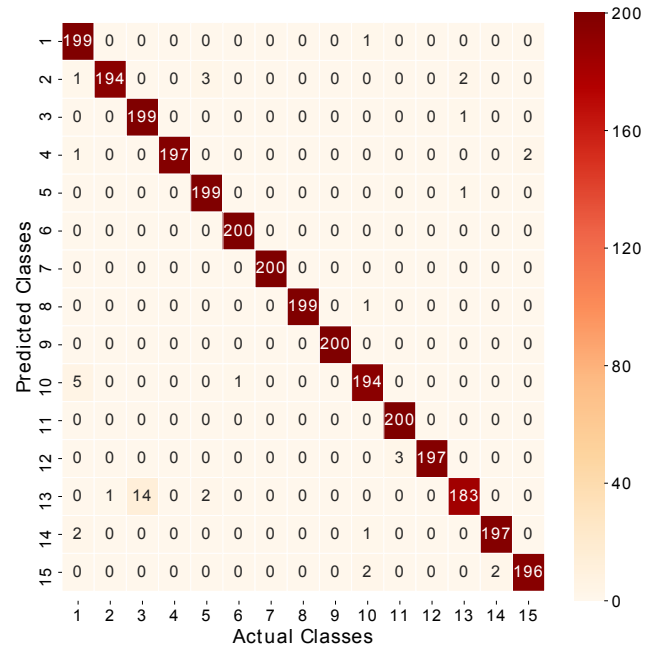


Fig. 7 – Confusion matrix for the merged model. Diagonal elements represent true positives, and off-diagonal elements represent the confusion between the classes.

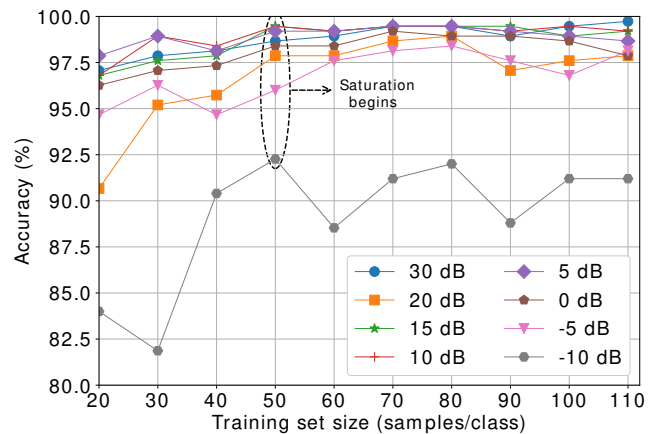


Fig. 8 – Classification accuracy as a function of training set size. Models give reasonable accuracies even with very low training data sizes. Saturation begins after 50 samples/class. We used 75 samples/class throughout this work.

rogram model at the same SNR level without denoising. Also note that, CNN models optimized using proposed denoising technique perform substantially better than both time-series image models and the models in [26], where conventional ML techniques are used at the latter, for every SNR level. For example, classification accuracies reported in [26] range from 40% to 98% , whereas CNN models trained on spectrograms with denoising range between 99.5% and 100% for SNR levels from 0 dB to 30 dB .

5.3.2 Model with a merged training set

Even though the models trained with different single-SNR data sets give satisfactory results, this approach comes with a practical difficulty. We can use these models only if

we can measure the SNR of a received signal prior to classification. In order to get rid of this requirement and also to save time, we merged the training sets of different SNR levels to create a more generalized model. The truncation cut-off that gives the smallest average loss was found to be -10 dB/Hz, therefore we merged all the images for eight SNR levels denoised at this cut-off threshold. Training and test sets are eight times greater than those of the single SNR sets. A classification accuracy of 98.8% (across all SNR levels) is achieved when using this model.

A confusion matrix of the merged model is given in Fig. 7. The major deficiency of the model is observed at (13, 3), where 14 out of 200 test data which belongs to class 3 is predicted as class 13. These two controllers belong to the same company and both their time-series plots and spectrograms show high virtual resemblance.

We also tested the merged model with images at intermediate SNR levels ranging from -12 dB to 22 dB with increments of 5 dB. We used 30 images for each class at each SNR, which add up to 3600 images, all previously unseen to the classifier, to test the model. Our model gives 96.9% accuracy, as shown in Table 4. In the case when we exclude the test data at -12 dB, the accuracy of the model increases up to 99.3%, which indicates that almost all the misclassification is associated with this particular SNR level.

5.4 Classification accuracy vs. training set size

There are popular CNN models in the literature that can be implemented to a wide variety of image classification problems via transfer learning, e.g., VGG16 or InceptionV3. These models have abundant hidden layers and have been trained over enormous data sets. Other than these models, it is more customary to come across CNN models that are deeper and trained on larger data sets in the literature. If the problem in hand is to accurately classify images of miscellaneous objects, e.g., humans, animals or cars, then a deep model with a very large number of training set should be required. This is because these images have more diversity in terms of position, angle, ambiance, lightning, etc. However, in our model, the set of images that we classify are generated by the well-defined methods that use the outputs of quite robust electronic circuitry. Thus, proposed models reach very high accuracy with as low as 100 training samples per class.

To better explain the sufficiency of a low number of samples for this particular problem, we examined the dependence of accuracy to the training data set size. Fig. 8 shows the accuracy of the classifier with respect to sample size per class for different SNR levels. In these simulations, the same models that are optimized for 100 samples per class are used for all cases. Note that, in this figure, x-axis denotes the size of the training sets only. We did not shrink the validation set while we tune the training set sizes. All models have been validated with a test

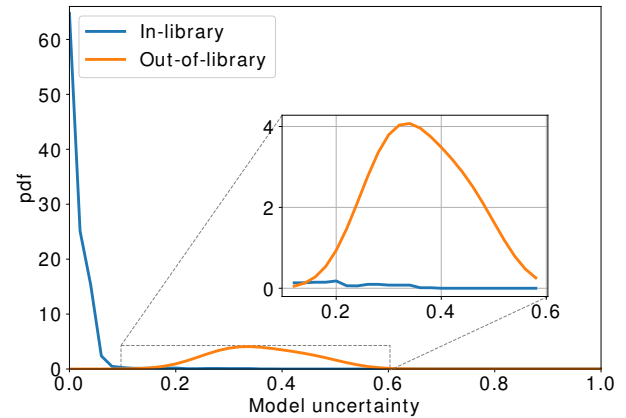


Fig. 9 – Pdfs of the model uncertainty for in-library and out-of-library UAV classes. The model predicts in-library signals with high certainty. The prediction uncertainty increases when the model encounters an out-of-library controller.

set of 25 samples per class unseen by the models before. Here it is seen that for small training set sizes, classification accuracy decreases as expected. After roughly 50 samples/class, the accuracy reaches saturation and begins to fluctuate. On the other hand, we see that the created models give reasonable accuracy even for a training set size of as low as 20 samples per class, which is because these samples are created by devices that have a high level of consistency.

For the practicality of the proposed system, the RF signal database should be updated as new products are introduced to the market. This also requires retraining of the CNN models, and hence fast training algorithms are needed. However, as explained above, the proposed system only requires a limited amount of training data, which in turn makes it a promising solution.

5.5 Out-of-library UAV controller signals

Finally, we investigate the behavior of the proposed algorithm when the receiver captures an out-of-library UAV controller signal. To do that, we tested our optimized CNN-based classifier for 40 signals from a Hubsan H501S X4 drone and compared the estimated probability distribution functions (pdfs) of the prediction uncertainty with those of the in-library test signals in Fig. 9. The output layer of the trained model gives a set of predictions for an incoming signal, where each element of the set corresponds to the estimated probability of that signal belonging to a particular class. A final decision on the class of the test signal is made based on the maximum probability, p_{\max} , in this set. We define the model uncertainty in Fig. 9 as $(1 - p_{\max})$.

We observe that the two classes (i.e., in-library and out-of-library UAVs) are well separated in terms of the model uncertainty associated with each of them, and out-of-library UAV signals can be easily identified by a simple thresholding mechanism. The threshold can be placed

based on the system requirements, i.e., the desired classification performance and false alarm rate. We recognize that a complete consideration of out-of-library classification requires adding out-of-library data in the training set or adaptation of open set recognition by introducing an OpenMax layer, which estimates the probability of an input being from an unknown class [38]. On the other hand, our proposed model gives very low model uncertainty for in-library signals, and this therefore still provides a reasonable solution described above to detect the out-of-library drones in practice.

6. CONCLUSION

In this study, we proposed a system that uses drone controller RF signals to classify drones of different makes and models for a wide variety of SNRs. We used CNN classifiers with two different sources for training the models: time-series images, and spectrogram images. We showed that the CNN model using the spectrogram images is more resilient to noise when compared with the time-series images based model. The proposed method that uses a merged training set of RF signals at different SNR levels along with the proposed denoising mechanism was shown to be effective for UAV classification even at SNRs not directly considered by the trained model. We also explored classification performance against training set size and showed that reasonable classification accuracy can still be obtained with limited training data. Consequently, adding new classes to the model (e.g., to include data from newly released drones) does not entail a high computation cost. Finally, we examined the model behavior with in-library and out-of-library drone signals and concluded that the proposed model shows a good performance identifying drones from an unknown class. Our future work includes comparing our results with federated learning techniques, and testing of the proposed CNN-based UAV classification technique at a larger scale, such as using the AERPAW experimental platform at NC State University.

ACKNOWLEDGMENT

This work has been supported in part by NASA under the Federal Award ID number NNX17AJ94A. The authors would like to thank Martins Ezuma at NC State University for providing the drone controller RF data set used in this study.

REFERENCES

[1] Hazim Shakhatreh, Ahmad H Sawalmeh, Ala Al-Fuqaha, Zuocho Dou, Eyad Almaita, Issa Khalil, Noor Shamsiah Othman, Abdallah Khreishah, and Mohsen Guizani. "Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges". In: *IEEE Access* 7 (2019), pp. 48572–48634.

[2] Dimas Pristovani Riananda, Galih Nugraha, Harish Mahatma Putra, Muhammad Lukman Baidhowi, and Riza Alaudin Syah. "Smart pulley workflow in delivery drone for goods transportation". In: *AIP Conference Proceedings*. Vol. 2226. 1. 2020, p. 060010.

[3] Connie A Lin, Karishma Shah, Lt Col Cherie Mauntel, and Sachin A Shah. "Drone delivery of medications: Review of the landscape and legal considerations". In: *The Bulletin of the American Society of Hospital Pharmacists* 75.3 (2018), pp. 153–158.

[4] Vinay Chamola, Vikas Hassija, Vatsal Gupta, and Mohsen Guizani. "A Comprehensive Review of the COVID-19 Pandemic and the Role of IoT, Drones, AI, Blockchain, and 5G in Managing its Impact". In: *IEEE Access* 8 (2020), pp. 90225–90265.

[5] Walid Saad, Mehdi Bennis, and Mingzhe Chen. "A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems". In: *IEEE Network* 34.3 (2020), pp. 134–142. DOI: 10.1109/MNET.001.1900287.

[6] M. Ritchie, F. Fioranelli, H. Griffiths, and B. Torvik. "Micro-drone RCS analysis". In: *Proc. IEEE Radar Conf. Johannesburg, South Africa, Oct. 2015*, pp. 452–456.

[7] Ismail Guvenc, Farshad Koohifar, Simran Singh, Mihail L Sichertiu, and David Matolak. "Detection, tracking, and interdiction for amateur drones". In: *IEEE Commun. Mag.* 56.4 (2018), pp. 75–81.

[8] Crown Consulting. *NASA Urban Air Mobility (UAM) Market Study*. URL: <https://ntrs.nasa.gov/citations/20190026762>.

[9] Federal Aviation Agency. *Proposed Rule on Remote Identification of Unmanned Aircraft Systems*.

[10] H. Zhang, C. Cao, L. Xu, and T. A. Gulliver. "A UAV Detection Algorithm Based on an Artificial Neural Network". In: *IEEE Access* 6 (May 2018), pp. 24720–24728.

[11] W. Zhou, L. Wang, B. Lu, N. Jin, L. Guo, J. Liu, H. Sun, and H. Liu. "Unmanned Aerial Vehicle Detection Based on Channel State Information". In: *Proc. IEEE Int. Conf. Sensing Commun. Netw. (SECON)*. Hong Kong, China, June 2018, pp. 1–5.

[12] M. Z. Anwar, Z. Kaleem, and A. Jamalipour. "Machine Learning Inspired Sound-Based Amateur Drone Detection for Public Safety Applications". In: *IEEE Trans. Veh. Technol.* 68.3 (Jan. 2019), pp. 2526–2534.

[13] Y. Seo, B. Jang, and S. Im. "Drone Detection Using Convolutional Neural Networks with Acoustic STFT Features". In: *Proc. IEEE Int. Conf. Advanced Video Signal Based Surveillance (AVSS)*. Nov. 2018, pp. 1–6.

- [14] V. Thai, W. Zhong, T. Pham, S. Alam, and V. Duong. "Detection, Tracking and Classification of Aircraft and Drones in Digital Towers Using Machine Learning on Motion Patterns". In: *Proc. Integrated Commun. Navig. Surveillance Conf. (ICNS)*. Herndon, VA, Apr. 2019, pp. 1–8.
- [15] M. Saqib, S. Daud Khan, N. Sharma, and M. Blumenstein. "A study on detecting drones using deep convolutional neural networks". In: *Proc. IEEE Int. Conf. Advanced Video Signal Based Surveillance (AVSS)*. Lecce, Italy, Aug. 2017, pp. 1–5.
- [16] J. Ren and X. Jiang. "Regularized 2-D Complex-Log Spectral Analysis and Subspace Reliability Analysis of Micro-Doppler Signature for UAV Detection". In: *Pattern Recognit.* 69 (Mar. 2017), pp. 225–237.
- [17] M. Marco and G. Pinelli. "Classification of Drones with a Surveillance Radar Signal". In: *Proc. Int. Conf. Comput. Vision Syst.* Thessaloniki, Greece, Sept. 2019, pp. 723–733.
- [18] P. Zhang, L. Yang, G. Chen, and G. Li. "Classification of drones based on micro-Doppler signatures with dual-band radar sensors". In: *Proc. Progress Electromagn. Research Symp. (PIERS)*. Singapore, Singapore, Nov. 2017, pp. 638–643.
- [19] A. Huizing, M. Heiligers, B. Dekker, J. de Wit, L. Cifola, and R. Harmanny. "Deep Learning for Classification of Mini-UAVs Using Micro-Doppler Spectrograms in Cognitive Radar". In: *IEEE Trans. Aerosp. Electron. Syst.* 34.11 (Nov. 2019), pp. 46–56.
- [20] B. K. Kim, H. Kang, and S. Park. "Drone Classification Using Convolutional Neural Networks With Merged Doppler Images". In: *IEEE Geosci. Remote Sens. Lett.* 14.1 (Jan. 2017), pp. 38–42.
- [21] B. Oh, X. Guo, F. Wan, K. Toh, and Z. Lin. "Micro-Doppler Mini-UAV Classification Using Empirical-Mode Decomposition Features". In: *IEEE Geosci. Remote Sens. Lett.* 15.2 (Feb. 2018), pp. 227–231.
- [22] P. Molchanov, K. Egiazarian, J. Astola, R. I. A. Harmanny, and J. J. M. de Wit. "Classification of small UAVs and birds by micro-Doppler signatures". In: *Proc. Eur. Radar Conf.* Nuremberg, Germany, Oct. 2013, pp. 172–175.
- [23] L. Wang, J. Tang, and Q. Liao. "A Study on Radar Target Detection Based on Deep Neural Networks". In: *IEEE Sens. Lett.* 3.3 (Jan. 2019), pp. 1–4.
- [24] A. Alipour-Fanid, M. Dabaghchian, N. Wang, P. Wang, L. Zhao, and K. Zeng. "Machine Learning-Based Delay-Aware UAV Detection and Operation Mode Identification Over Encrypted Wi-Fi Traffic". In: *IEEE Trans. Inf. Forensics Security* 15 (Dec. 2019), pp. 2346–2360.
- [25] N. Regev, I. Yoffe, and D. Wulich. "Classification of single and multi propelled miniature drones using multilayer perceptron artificial neural network". In: *Proc. Int. Conf. Radar Syst.* Belfast, UK, Jan. 2017, pp. 1–5.
- [26] M. Ezuma, F. Erden, C. Kumar Anjinappa, O. Ozdemir, and I. Guvenc. "Detection and Classification of UAVs Using RF Fingerprints in the Presence of Wi-Fi and Bluetooth Interference". In: *IEEE Open J. Commun. Soc.* 1 (Nov. 2019), pp. 60–76.
- [27] Martins Ezuma; Fatih Erden; Chethan K. Anjinappa; Ozgur Ozdemir; Ismail Guvenc. "Drone Remote Controller RF Signal Dataset". In: *IEEE Dataport*, 2020. DOI: 10.21227/ss99-8d56. URL: <https://dx.doi.org/10.21227/ss99-8d56>.
- [28] Z. Shi, X. Chang, C. Yang, Z. Wu, and J. Wu. "An Acoustic-Based Surveillance System for Amateur Drones Detection and Localization". In: *IEEE Trans. Veh. Technol.* 69.3 (Mar. 2020), pp. 2731–2739.
- [29] C. Aker and S. Kalkan. "Using deep networks for drone detection". In: *Proc. IEEE Int. Conf. Advanced Video Signal Based Surveillance (AVSS)*. Lecce, Italy, Aug. 2017, pp. 1–6.
- [30] Y. Zhao and Y. Su. "The Extraction of Micro-Doppler Signal With EMD Algorithm for Radar-Based Small UAVs' Detection". In: *IEEE Trans. Instrum. Meas.* 69.3 (Apr. 2020), pp. 929–940.
- [31] B. Choi and D. Oh. "Classification of Drone Type Using Deep Convolutional Neural Networks Based on Micro-Doppler Simulation". In: *Proc. Int. Symp. Antennas Propag. (ISAP)*. Busan, South Korea, Oct. 2018, pp. 1–2.
- [32] A. Shoufan, H. M. Al-Angari, M. F. A. Sheikh, and E. Damiani. "Drone Pilot Identification by Classifying Radio-Control Signals". In: *IEEE Trans. Inf. Forensics Security* 13.10 (Mar. 2018), pp. 2439–2447.
- [33] Roberto Brunelli. *Template Matching Techniques in Computer Vision: Theory and Practice*. John Wiley & Sons, Ltd, 2009. ISBN: 9780470744055.
- [34] P. Sysel and Z. Smékal. "Enhanced estimation of power spectral density of noise using the wavelet transform". In: *Pers. Wireless Commun.* Boston, MA: Springer US, 2007, pp. 521–532. ISBN: 978-0-387-74159-8.
- [35] Francisco-Cervantes Cervantes, Jesus Gonzalez-Trejo, Cesar Real-Ramirez, and Hoyos-Reyes Luis. "Fractal dimension algorithms and their application to time series associated with natural phenomena". In: *J. Phys.: Conf. Ser.* 475 (Nov. 2013).
- [36] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 2019.

[37] A. Bhandari, H. Yin, Y. Liu, W. Yao, and L. Zhan. "Real-Time Signal-to-Noise Ratio Estimation by Universal Grid Analyzer". In: *Proc. Int. Conf. Smart Grid Sync. Meas. Anal. (SGSMA)*. College Station, TX, Aug. 2019, pp. 1–6.

[38] A. Bendale and T. E. Boulton. "Towards Open Set Deep Networks". In: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Las Vegas, NV, Dec. 2016, pp. 1563–1572.

AUTHORS



Ender Ozturk received his B.S. and M.S. degrees from Bilkent University, Ankara, Turkey, in 2005 and 2008, respectively, and the Ph.D. degree from Hacettepe University, Ankara, Turkey, in 2018, all in electrical and electronics engineering. He held different positions in industry as an engineer, senior

engineer and team leader. He is currently working as a Postdoctoral Research Scholar at North Carolina State University.



Fatih Erden received the B.S. and M.S. degrees from Bilkent University, Ankara, Turkey, in 2007 and 2009, respectively, and the Ph.D. degree from Hacettepe University, Ankara, Turkey, in 2015, all in electrical and electronics engineering.

From 2015 to 2016, he was an Assistant Professor at Atilim University, Ankara, Turkey. He is now working as a research associate at North Carolina State University. His research interests include signal and image processing, machine learning, time series analysis, mmWave communications, and UAVs.



Ismail Guvenc is Professor at North Carolina State University. His recent research interests include 5G wireless networks, UAV communications, and heterogeneous networks. He has published more than 200 conference/journal papers, several standardization contributions, three books, and over

30 U.S. patents. He is a recipient of the 2019 R. Ray Bennett Faculty Fellow Award, 2016 FIU COE Faculty Research Award, 2015 NSF CAREER Award, 2014 Ralph E. Powe Junior Faculty Award, and 2006 USF Outstanding Dissertation Award, and he is a senior member of the National Academy of Inventors.