Readme document for drone RF database acquisition to be used in detection and identification applications [1]

The additional material in this file is provided by the authors of [1] to assist the reader to better understand the developed database and how to use it for drone detection and identification applications. The drone's raw RF database can be obtained using the OneDrive link in [2]. In addition, the developed LabVIEW, MATLAB, and Python implementations can be found in the permanent GitHub link in [3].

The MATLAB and Python scripts listed below, supplied within the GitHub repository, are used to produce results and supporting figures illustrated in the paper. Kindly run the following scripts in sequence to regenerate all our results from scratch.

Main Scripts:

Main_1_Data_aggregation.m

- Description: This script produces MATLAB formatted data that will used later by *Main_2_Data_labeling.m* and *Demo_3_Analysis.m* scripts.
- Process: It loads the raw RF data, re-segments it by a factor of 100, transforms it to the frequency domain using 2048 samples, and aggregate the results in a MATLAB data format using the following file names: *00000.mat, 10000.mat, 10001.mat, 10010.mat, 10011.mat, 10100.mat, 10101.mat, 10110.mat, 10111.mat,* and *11000.mat*.

Main_2_Data_labeling.m

- Description: This script produces csv formatted data that will used later by *Classification.py*.
- Process: It loads the MATLAB data files generated by *Main_1_Data_aggregation.m*, normalizes them, concatenates them in a matrix, adds classification labels, and save them in a csv file named *RF_Data.csv*.

Classification.py

- Description: This script produces csv formatted data that will used later by *Demo_4_Classification.m*.
- Process: It loads the csv data file generated by *Main_2_Data_labeling.m*, extracts RF data and labels, and then trains three deep neural networks (DNNs) to classify the RF data. The DNNs are cross-validated using a stratified 10-fold cross-validation method. Predicted labels for each fold are computed and saved in a csv format data file.

Demo Scripts:

Demo_1_Database_details.m

- Description: This script produces the results that are depicted in Table 4 of the paper in [1].

- Process: It calculates the number of acquired raw samples and segments for each class at each experiment level. After that, it displays the computed details along with the class size ratios to check for class imbalance issues.

Demo_2_Snippets.m

- Description: This script produces the results that are depicted in Figure 11 of the paper in [1].
- Process: It loads segment number 5 of the acquired RF background activities, segment number 10 of the acquired Bebop RF signals when flying and video recording, and lastly, segment number 7 of the acquired Phantom RF signals when on and connected. After that, it normalizes the amplitude of each segment and plots the normalized RF signals.

Demo_3_Analysis.m

- Description: This script produces the results that are depicted in Figure 12 of the paper in [1].
- Process: It loads the MATLAB data files generated by *Main_1_Data_aggregation.m*. After that, it averages all instances of each spectra and smooths them using a 10-point moving average filter. Then, it plots the averaged smoothed spectra along with their boxplots. Finally, it asks the user whether to save the results in a PDF format image. On can generate the results of each column in Figure 12 by altering the "*opt*" parameters from 1 to 2, and finally to 3.

Demo_4_Classification.m

- Description: This script produces the results that are depicted in Figure 13 of the paper in [1].
- Process: It loads the csv data files generated by *Classification.py*. After that, it concatenates all predicted labels and computes their respective confusion matrix using the MATLAB function *plotconfusion_mod*. Finally, it asks the user whether to save the results in a PDF format image. On can generate the results of each column in Figure 13 by altering the "*opt*" parameters from 1 to 2, and finally to 3.

Functions:

plotconfusion_mod

- Description: This function is a modified version of the MATLAB built-in function *plotconfusion*.
- Process: It computes the conventional confusion matrix but using custom fonts and appended with extra results, i.e. F1 score. See Section 3.3.4 in [1] for more details.

References

[1] M. Al-Sa'd et al. "RF-based drone detection and identification using deep learning approaches: an initiative towards a large open source drone database", 2019.

[2] Drone's raw RF database: https://tutfi-my.sharepoint.com/:f:/g/personal/mohammad_al-sad_tut_fi/EtEuLLSAO9RO1efSh8Nzb18BswRNTo9vOUzE94rRjhrrxQ?e=n2H1je.

[3] M. F. Al-Sa’d, A. Al-Ali, A. Mohamed, T. Khattab, Drones acquired RF database along with the utilized software, Online (2019). URL: https://al-sad.github.io/DDI_Software/.