University of New Haven

Data Analysis and modeling of radar measurements for detecting of Unmanned Aerial Vehicles (UAVs)

DSCI-6051 Data Science Capstone Project

Project Report

Prepared By:

Srimanikanta Arjun Karimalammanavar

Graduate Student – Data Science

University of New Haven – sarju1@unh.newhaven.edu

With the guidance of:

Dr. Eric A. Dieckman, Ph.D.

Assistant Professor – Mechanical Engineering

University of New Haven – Edieckman@newhaven.edu

Divyansh P Patel

Graduate student – Computer Science

University of New Haven – dpate43@unh.newhaven.edu

Prepared for:

Dr. Vahid Behzadan, Ph.D.

Assistant Professor – Data Science

University of New Haven – Vbehzadan@newhaven.edu

Introduction

The use of unmanned aerial vehicles (UAVs) has grown dramatically in a variety of applications such as surveillance, air traffic management, and civil and commercial demands. In this project, we address this issue by trying to build a machine learning model to classify drones based on their micro-doppler shift captured in the heat maps

Radar

The AWR1642 device is an integrated single-chip FMCW radar sensor capable of operation in the 76- to 81-GHz band. The device is built with TI's low-power 45-nm RFCMOS process and enables unprecedented levels of integration in an extremely small form factor. The AWR1642 is an ideal solution for low-power, self-monitored, ultra-accurate radar systems in the automotive space.

The AWR1642 device is a self-contained FMCW radar sensor single-chip solution that simplifies the implementation of Automotive Radar sensors in the band of 76 to 81 GHz. It is built on TI's low-power 45-nm RFCMOS process, which enables a monolithic implementation of a 2TX, 4RX system with built-in PLL and A2D converters. It integrates the DSP subsystem, which contains TI's high-performance C674x DSP for the Radar Signal processing. The device includes an ARM R4F-based processor subsystem, which is responsible for radio configuration, control, and calibration. Simple programming model changes can enable a wide variety of sensor implementation (Short, Mid, Long) with the possibility of dynamic reconfiguration for implementing a multimode sensor. Additionally, the device is provided as a complete platform solution including TI reference designs, software drivers, sample configurations, API guides, and user documentation.

Data Collection

The collection of data for the drones was straightforward using the GUI provided by the dev.ti.com.

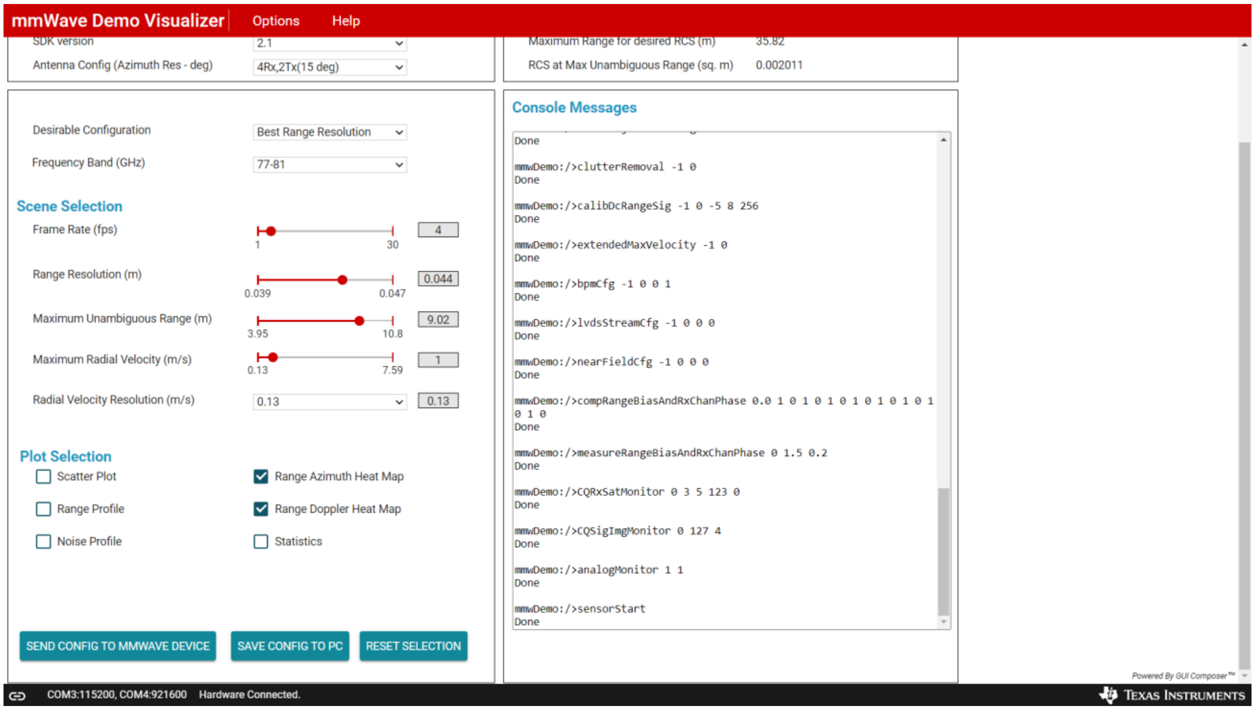


Fig: GUI to collect data

We get the heatmaps of the doppler-range using the interaction between this web-based GUI and the ports connected to the mmWave radar using the USB.

The drones used were:

1. Parrot Mambo Fly
2. Parrot Swing
3. DJI Tello

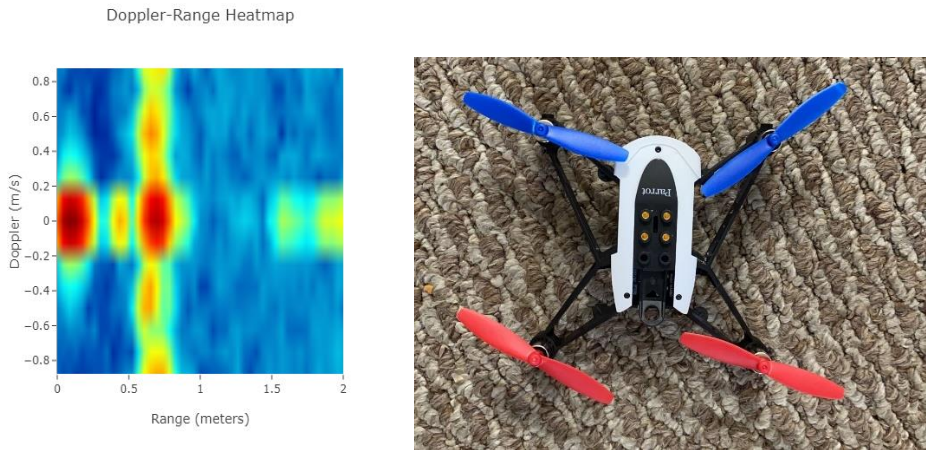


Fig: Heat map (Left) of Parrot Mambo Fly (Right)

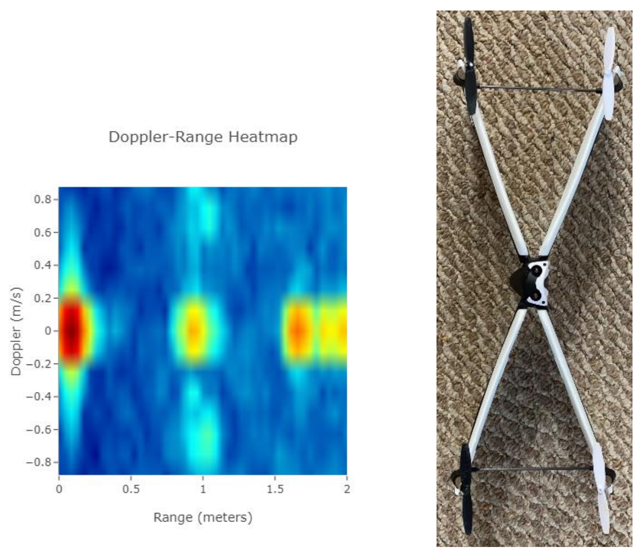


Fig: Heat map (Left) of Parrot Swing (Right)

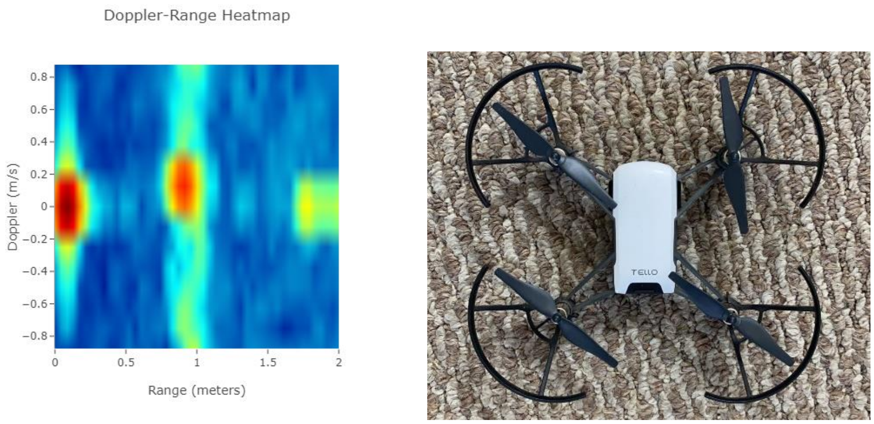


Fig: Heat map (Left) of DJI Tello (Right)

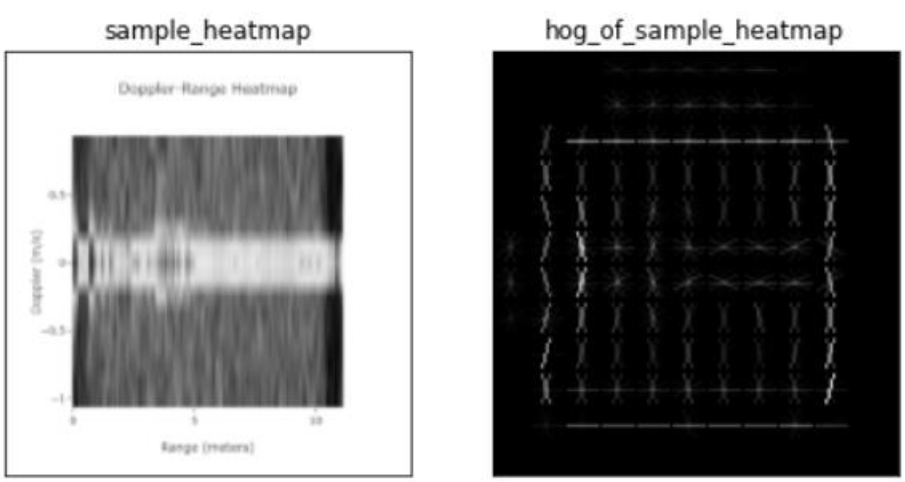
Models

Two models were implemented to this problem.

1. Stochastic Gradient Descent Classifier
2. Convolutional Neural Network

Stochastic Gradient Descent Classifier

1. The images were resized as width was made equal to the height of the image.
2. Next, we split our data into 80% for training and 20% for testing our model.
3. A well-defined approach to object recognition in Machine learning is usage of HOG-SVM which stands for Histogram of Oriented Gradients and Support Vector Machines.
4. HOGs are used for feature reduction, i.e., lowering the complexity of the problem while maintaining as much variation possible.



5. The above figure shows the HOG of a sample heatmap
6. The advantage of this method is reducing the number of features per image to less than ~10% of the original features.
7. We use BaseEstimator, TransformerMixin from sklearn to build two transformers:
    a. RGB2GrayTransformer
    b. HogTransformer
8. We use StandardScaler to scale features.
9. We then transform the train data using these transformers.
10. We define the SGD Classifier to fit the training data and train the model.
11. We repeat the transformation steps for test data and test our model using the predict method.
12. We have achieved an accuracy of 95%.

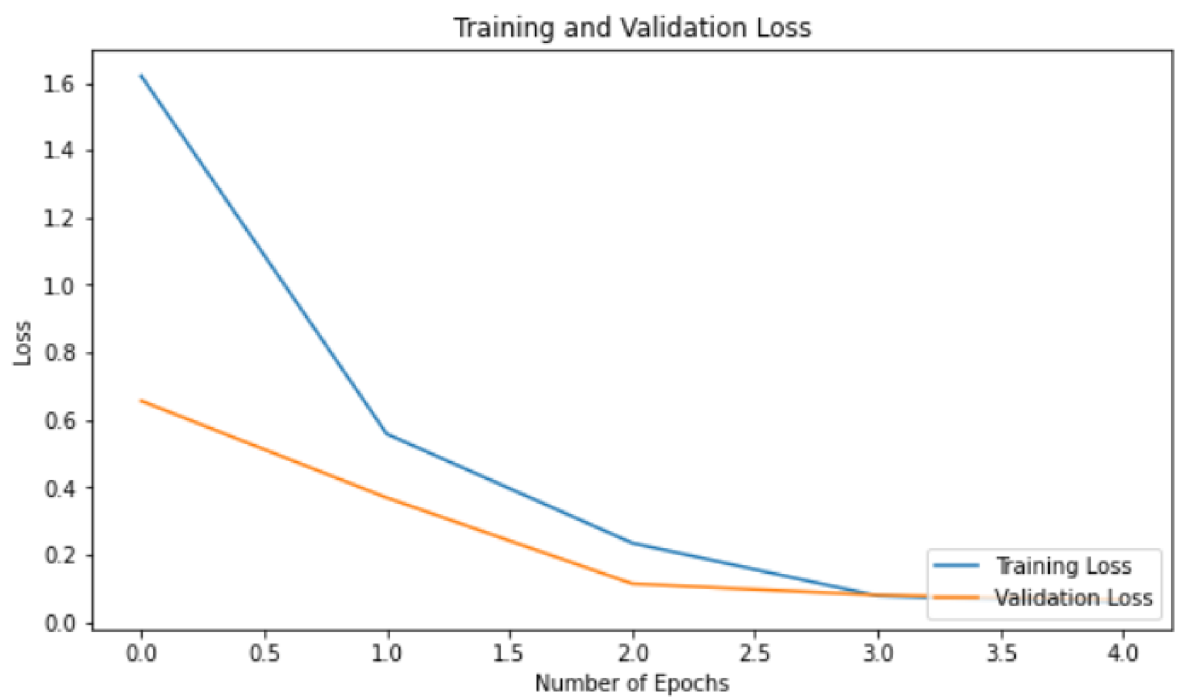
```
In [19]: from sklearn.metrics import confusion_matrix

         cmx = confusion_matrix(y_test, y_pred)
         cmx

Out[19]: array([[20,  0,  2],
                [ 0, 16,  0],
                [ 1,  0, 21]], dtype=int64)
```

13. From the confusion matrix, we can see that Parrot Mambo fly was misclassified as DJI Tello 2 times and DJI Tello being misclassified as Parrot Mambo Fly as 1 time.
14. This is because, both the drones resemble similar in shape, propeller orientation, etc.,.

Convolutional Neural Network

1. We have used 400 heatmaps from each drone for building our CNN.
2. We have split our dataset into 80% training and 20% validation for testing purposes.
3. We have built a CNN with 3 convolutional layers with a max pool layer in each of them. There’s a fully connected layer with 128 units on top of it that is activated by a relu activation function
4. We have chosen the “adam” optimizer and “SparseCategoricalCrossEntropy” loss function and have passed the “metrics” argument.
5. We have defined 5 epochs.
6. We have achieved an accuracy of 98.05% training accuracy and 98.76% validation accuracy at the end of 5 epochs.

7. Our loss has decreased significantly, from 3.2 to 0.07 for the training data and 0.6563 to 0.0665 for the validation data.



Limitations

1. The major limitation was the use of .png files as data as data compression when saving the .png file might lead to data loss.
2. The way to improve this is the use of numerical data from the radar.
3. With the numerical data, we can build our own histograms, extract features from them and build models which would prove to be more efficient.

Future Work

1. The next step is definitely the use of numerical data from the radar for the models.
2. The same models can be used or models like Decision tree, K-nearest neighbors, Random forest can be used.

Conclusions

In this project, we recorded heat maps of three different drones. Then we built a SGD classifier on the data using the HOG features of the input images and achieved an accuracy of 95%. We also tried a deep Convolutional Neural Network on the data and achieved an accuracy of 98%.

References

1. Dongsuk Park, Seungeui Lee, SengUk Park and Nojun Kwak. Radar-Spectrogram-Based UAV classification using Convolutional Neural Networks. Sensors 2021,21,210.
2. Passafiume, Marco; Rojhani, Neda; Collodi, Giovanni; Cidronali, Alessandro. 2021. "Modeling Small UAV Micro-Doppler Signature Using Millimeter-Wave FMCW Radar" *Electronics* 10, no. 6: 747. https://doi.org/10.3390/electronics10060747