Sensor Fusion for Drone Detection

Mohammed Aledhari*, Rehma Razzak*, Reza M. Parizi*, Gautam Srivastava‡
*College of Computing and Software Engineering, Kennesaw State University, GA, USA
‡Department of Mathematics and Computer Science, Brandon University, Manitoba, Canada

2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring) | 978-1-7281-8964-2/20/$31.00 ©2021 IEEE | DOI: 10.1109/VTC2021-Spring51267.2021.9448699

***Abstract*—With the rapid development of commercial drones, drone detection and classification have emerged and grown recently. Drone detection works to detect unmanned aerial vehicles (UAVs). Usually, systems for drone detection utilize a combination of one or more sensors and some methodology. Many unique technologies and methods are used to detect drones. However, each type of technology offers its benefits and limitations. Most approaches use computer vision or machine learning, but one methodology that has not been given much attention is Sensor Fusion. Sensor Fusion has less uncertainty than most methods, making it suitable for drone detection. In this paper, we propose an artificial neural network-based detection system that uses a deep neural network (DNN) to process the RF data and a convolutional neural network (CNN) to process image data. The features from CNNs and DNNs are concatenated and input into another DNN, which outputs a single prediction score of drone presence. Our model achieved a validation accuracy of 75% that shows the feasibility of a sensor fusion based technique for drone detection.**

***Index Terms*—DNN; CNN; multi-sensor; data fusion; drone; UAVs**

I. Introduction

The research area of drone detection and classification has emerged and grown recently. Drone detection works to detect unmanned aerial vehicles (UAVs). Fig. 1 presents an overview of various methods that are typically used for drone detection with their limitations evaluated on a scale of 1 to 5.

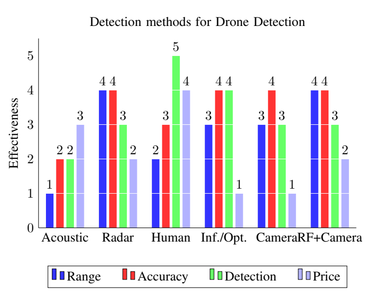

Fig. 1: A scale from 0 to 5 showing the effectiveness of various drone detection methods in comparison to a fusion approach of RF and camera detection [6].

Designing systems for drone detection is often easier said than done because many of these systems are prone to design issues that hamper their overall performance. Other issues these systems have encountered include weather, visibility, and false positives such as detection of birds or planes. To alleviate these drawbacks, alternative approaches have been implemented, such as using optical sensors. Optical sensors have been used for several types of object detection and have been successful in the detection of UAVs [1]. Similarly, another alternative has been to use radio frequency (RF) scanners. There have been several studies using radio frequency scanners to detect Micro Doppler signatures to perform drone detection and classification [6].

Ultimately, the existing methods over the classification and detection of drones mainly focused on a single sensor approach. However, there have been other methods used which involve the camera. The camera is commonly used in object detection, usually with DNNs to detect drones [6]. Even then, these methods have not returned optimal results and they also involve contain large quantities of data. As such, there are still concerns about performance, efficiency, and reliability between sub-optimal conditions. RF data has emerged as an approach for drone detection as well, with current implementations that use RF data involving Support Vector Machines (SVM), CNN, and DNN to accomplish classification [4].

Being able to accurately detect drones as efficiently as possible is crucial because due to the increase in technological advances with commercial Unmanned Aerial Vehicle (The terms such as drones, UAV, and Unmanned Aerial Vehicle are used equally through this paper), the danger of misuse and hostile use of drones are also on the rise. Some possible major concerns surrounding UAVs are interference with manned flight, unwanted trespassing in restricted locations, and the illegal transportation of regulated substances [7]. There are threats to privacy and security as well since most drones are easily equipped with a camera, which further increases the risk factors involved with privacy. Therefore, there is a need for a robust system capable of detecting UAVs in such areas.

A. Research Problem

The detection of drones through single sensor computer vision methods brought forth challenges with suboptimal visual conditions [1], malfunctions in hardware, and similarities between bio-life creatures. These suboptimal conditions increase the likelihood of false-positive results occurring [1]. The problem arises due to a constantly increasing number of commercial UAVs occupying the skies. These UAVs pose an extensive threat if they are found entering the airspace of an airport. This problem not only affects the safety of personnel and passengers at the airport of origin but also affects the operations of any airport. On average, Hartsfield-Jackson International Airport sees 2,500 flights take off per day. If runway operations were only to be halted for thirty minutes, approximately 52 flights potentially carrying thousands of passengers would be delayed [10]. This bears serious consequences such as serious injury, extensive damage to third-party assets, and a series of delays that spread like a virus. Additionally, it is crucial to determine which classifier approaches for classifying drones is most optimal, because not all classifiers may be suitable for drone detection; Some classifiers take more time than other methods for drone detection, which is not always affordable when drones usually fly at high speeds. Therefore, the chosen approaches have to account for that too.

B. *Purpose of the Study*

The goal of this paper is to propose a solution to these problems. The nature of this topic dictates the use of both newer methodologies and open-source comprehensive datasets to solve our detection problem as efficiently as possible while achieving higher accuracy. As technological advances to drones gradually increase, the danger of misuse rises due to the increased availability of UAVs [7]. Alternatives to these automotive drone detection systems often have a higher cost of maintaining the manual systems or have a significant decrease in performance of drone classification and detection in sub-optimal visual conditions [6]. Proposing a solution to these possible issues not only decreases the cost of alternative methods but also increases the accuracy and precision of drone-based classification and detection. More extensive research and implementation of sensor fusion will enable drones to be detected with higher accuracy much more quickly.

C. *Contribution*

Most studies fail to specify the type of acquisition device, the drone type, the detection range, or the employed dataset when attempting to solve the problem of drone-detection. Previous studies implemented multiple approaches, such as machine learning for example, but these approaches can be more demanding. Additionally, several aspects may be compromised, which affects the accuracy of drone detection. Some aspects that may be affected are visibility and sensitivity. The present work is designed to be a more efficient solution for drone detection by implementing a model that utilizes sensor fusion in combination with multiple datasets that contain drones in flight. The main contribution of this paper includes a model that utilizes sensor fusion between RF and images. The details of our model are given later on in the paper, but since our work is heavily implementation based, our contribution helps increase the motivation towards starting and implementing this study. Additionally, our work also helps increase awareness about Sensor Fusion, a technique that has been given little attention concerning drone detection. With our work, we are contributing more knowledge of Sensor Fusion's applicability in drone detection. This is crucial because currently, literature regarding Sensor Fusion is limited.

II. Related works

The increasing presence of UAVs within society brings issues that will increase the demand for drone-based classification and detection. Reference [7] addresses the technological similarities and differences between various drone models but provides insight over the variations between trends in drone purchases, size, and model over the years.

Reference [1] goes over how a deep neural network can be used with stationary cameras to detect differentiation between backgrounds and new objects. Additionally, [9] discusses the applicability of machine learning for drone detection. Machine learning can be suitable for drone detection due to its ability to recognize patterns.

Reference [2] has provided a Radio Frequency dataset due to the small amount of available relevant and open-source datasets for the various modes of several different drone types recording the RF emitted. With the use of a Deep Neural Network (DNN), the article has also included relevant metrics results over performance results over the classification of drones from the RF emitted during these modes to assist in showing the reliability of RF-based detections.

In addition to the metrics from the RF dataset analyzed in [2], the actual RF dataset in addition to some python and MATLAB programs can be utilized to view the relevant data in a similar way that [2] used. Reference [3] provided four different samples of RF recordings, three from different drone types and one consisting of just background noise to differentiate between the different drone recordings [3].

There have also been techniques that have used other sensors, mainly camera and Lidar sensors. For example, the work by [8] details the process of designing and implementing an automatic MSDDS (multi-sensor drone detection system) using thermal and acoustic sensors. The authors use not only the typical video and audio sensors, but they also include a thermal infrared camera. The authors also discuss both limitations and possibilities of designing and implementing an MSDDS system based on machine learning techniques. Furthermore, the authors also managed to make a video dataset, containing 650 infrared and visible videos of drones, airplanes, and helicopters; The infrared videos have a resolution of 320×256 pixels and visible videos have a resolution of 640×512 pixels. The authors' dataset also includes an audio dataset of drones, helicopters, and background noise; there are 90 audio clips total. For the experiment, the authors focused solely on performance. Additionally, the authors increased the number of target classes and used three different consumer-grade drones and organized performance results based on distance; Distance was categorized as follows: close, medium, and distance. The authors' evaluation focused on measuring performance in terms of precision, recall, and f1-score. Upon examining the results the authors noted that common false alarms in drone detection were small clouds and edges of large clouds that were lit up via the sun. The authors also noted that results were less optimal when sensor to target distance increased. Interestingly, the article mentions that sensor fusion was also used; unfortunately, the process of how the sensor fusion method was utilized is hard to follow, making it difficult to understand.

Regarding sensor fusion techniques [5] is one of the few that do discuss sensor fusion. However, the authors focus on multisensor data fusion, which is related to sensor fusion. With sensor data fusion, the authors explain that this deals with combining observations from various sensors to achieve optimal detection of targets. The authors then explain that Sensor data fusion consists of three major tasks: data association, target detection, and target localization. The data association task identifies which observations or from different sensors belong to the same target and which are false positives. In the target detection task, it determines how many targets are detected and which sensor(s) contribute to the detection of such targets. For target localization, a fusion algorithm is implemented for location and uses the provided data. Overall, the authors do an excellent job of discussing the role of sensor data fusion in UAV detection.

While many of these works are detailed, not all of them discuss or implement sensor fusion. Additionally, not all the articles mention what metrics are used for their experiments. Some of these articles' process on how they incorporate sensor fusion is unclear and harder to follow in comparison to our work. Additionally, given that so few papers discuss the results of implementing sensor fusion, we were unable to find a suitable base for comparison. It should also be noted in comparison to other works, we incorporate more evaluation metrics to provide a strong analysis of our results. While it would have been optimal to perform fair and unbiased comparisons against any existing works, we were unable to do so for the following reasons: (1) the limited literature regarding sensor fusion, and (2) lack of availability of code to be used by researchers. Additionally, many of the literary works we examined were out of scope for our purposes, with some works focusing on specific use-cases for drone detection and other works addressing different issues relating to the drone itself (e.g. physical hardware, drifting issues), and therefore not applicable for comparison. The lack of availability of code for topics such as this is a significant detriment due to the lack of publically available reference datasets for drone detection, which makes implementation and comparison difficult to do.

III. Techniques Used

The main technique used in this study is Sensor Fusion. We also use a camera and an RF receiver, which are combined to improve drone detection. As such, these signals must be preprocessed separately and combined to produce a detection result. The processing of these two inputs is carried out with some data preprocessing upfront which resamples the RF signals and creates the images into features. In our case, sensor fusion is applicable since we use multiple datasets that have different formats and types to detect the drone with high confidence.

IV. Datasets Used

The datasets used are the RF Dataset, the Image Dataset, and a combined dataset. These are discussed in more detail in the following subsections. We collected and labeled the datasets and appended them to TensorFlow as a library. Then used one available drone dataset that we found. For the RF dataset, we used the SVM classifier; So no layers or neural networks are involved.

Regarding how we collected images and paired them with RF data, First we collected images and labels of approximately 4000 drones, birds, helicopters, and airplanes, some of which were created from scratch. Then, we created Python scripts to transfer the JPG and XML files associated with the datasets to their proper location; We also created a program to convert the XML files to a CSV file (a table containing all detailed data for a dataset), and another program to convert each CSV to a TFRecord for TensorFlow to interpret the dataset, as well as a handful of other scripts used to clean up our dataset. Once completing all required conversions, we used the trained the datasets using multiple pre-trained models. This allowed us to obtain the best possible results regarding what the computer perceives is and is not a drone.

As such, our system is implemented using a dual-camera system and a Radio Frequency (RF) analyzer for increased accuracy and precision. This consists of a static broad-angle camera and a lower-angle camera that is mounted on a revolving frame. In our system, we propose an RF analyzer that is also attached to the rotating turret. The cameras will be hooked up to a computer allowing for real-time object detection, object tracking, and image classification. Using Python, OpenCV, and TensorFlow in conjunction with the pre-trained models, we have trained the computer to identify drones, birds, helicopters, and airplanes to increase its ability to tell one apart from the other. As a result, confidence is improved and fewer false positives occur.

We used Python, OpenCV, and TensorFlow to find the drone within a frame. First, we convert the frame to grayscale to decrease the computational resources required. Once the frame is in grayscale, we used a matrix of pixels to traverse through subsections of the frame to find the features (edges) of a drone. The features it looks for are features we trained the computer to specifically focus on concerning drones. We accomplished this task using a dataset containing hundreds of unique drone images. To identify whether the frequency belongs to drones or not, we must consider the frequencies drones use. Drones typically operate at two standard frequencies: 2.4GHz or 5GHz. Drones operating on a 5GHz frequency are easier to detect in theory. This is because many home devices such as Wi-Fi, garage openers, and cordless cell phones operate on a 2.4GHz frequency. Also, drones operating on 5GHz frequencies have twenty-three channels while drones operating on 2.4GHz have eleven, thus contributing to more significant interference using the 2.4GHz frequency.

Using an RF analyzer, we detect the frequencies generated between the drone and the controller, which will allow us to determine if what is being detected is a drone rather than a bird or a plane. It does this by comparing the frequencies it detects to a recorded drone frequency database that depicts radio waves caused by different drone functions such as flying and recording versus hovering.

A. The RF Dataset

Because there is a limited amount of data available in terms of RF signals of drones, we used an open-source, comprehensive dataset, which is described in [3]. That dataset suited our needs the best, so we used it for our detection problem. The dataset is composed of 227 segments, of three different drones collected using RF receivers. Each segment is composed of a high and low band of 2.5GHz frequency. There are some limitations with only analyzing 2.5GHz, as some commercially available drones also use the 5GHz frequency band to communicate. However, most current drones still use the 2.4GHz band for remote control and video. Furthermore, each segment is composed of 10 million points for each of the higher and lower bands of frequency, which makes the size of the complete database around 40GB.

We could not use the complete dataset for this study because the background RF noise samples only amount to 40 distinct segments. So, to maintain a balanced dataset that will not skew results, 40 additional segments were chosen from the dataset. In this aspect, we were prioritizing modes where the drone was flying and recording video. The complete dataset was composed of 80 samples evenly split with background RF noise and drone RF signals, which should provide a good representation of both classes for our proposed model.

The selected 80 samples (with each sample consisting of 20 million points) amounted to a data-size of 12GB. Processing and building a model to analyze such a dataset would be computationally very expensive. Therefore, several preprocessing steps were performed to reduce the dimension of the data. As a result, the dataset was resampled using similar techniques to those described in [3]. The 20-million-point RF signal for both the high and low band was resampled to 4882 points using the same bin size of 2048. The resampling process used the SciPy signal Python library and the Fourier transform technique.

The resampling process removes the second and third groups of a certain amount of elements. The preprocessing reduces the dimension of a single segment from 20 million to 9764 samples. This greatly boosts the performance of a model processing these samples. Although the resampling process enables the reduction of data, certain features or trends may be lost in this process. It should be noted that the RF signals work all the time for the drone. The images depend on our camera, so there is no connection. Essentially, we are detecting images using different resources.

Figs. 2 and 3 show the original and resampled RF signals respectively. The figures show an example where a drone is present and how the necessary features that can be used to detect the drone are maintained. Despite some edge cases where the signal does not retain the information, the transformation seems to deal with RF signals with drones and background noise well. The reduced dimensionality also allows for a real-time model with a bit of upfront computational expense for the transform. Other methodologies kept the signal in the frequency domain [2] arguing that it provides a better representation. The signals were also normalized to be between -1 and 1 as a preprocessing step before it was used by the model. We chose to use -1 and 1 values to make the RF data easier to work with.

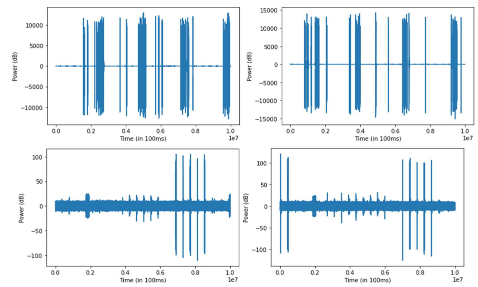

Fig. 2: Lower band of an RF segments composed of 10 million points plotted.

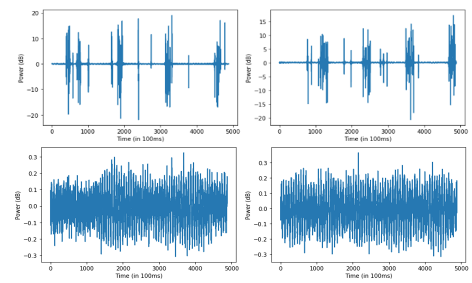

Fig. 3: Lower band of RF segments (4882 points plotted).

B. The Image Dataset

Unfortunately, a dataset that includes both images and RF data was not found. Ideally, such a dataset would provide the best assessment of a drone detection system capable of fusing the data from both sensors. Due to the lack of such a dataset, a simulated dataset was created. The images are collected independently from the RF dataset and later combined to use in the model. The images were filtered for quality manually and paired with the RF to create 80 samples. The image transformations were quite simple. First, the image was reduced to a single channel (grayscale), rescaled to 300×300 and normalized to be between 0 and 1. We chose to use 0 and 1 as parameters because we thought it would be good parameters for classes: 1 meaning there is a drone, and 0 representing the background. That way, it becomes clearer to understand whether drones or detected or not. The results of these operations are shown in Figs. 4 and 5. This resolution was chosen as it maintains enough quality to distinguish objects like a bird and a drone. The images were specifically picked such that objects like the drone and birds were recognizable. Something to note is that the image rescaling is taking into account the preservation of the aspect ratio, and it will not affect results because the CNN handles the rescaling of the images.

C. The Combined Dataset

The image and RF datasets were combined to produce the final dataset, which was composed of 80 samples. The size is

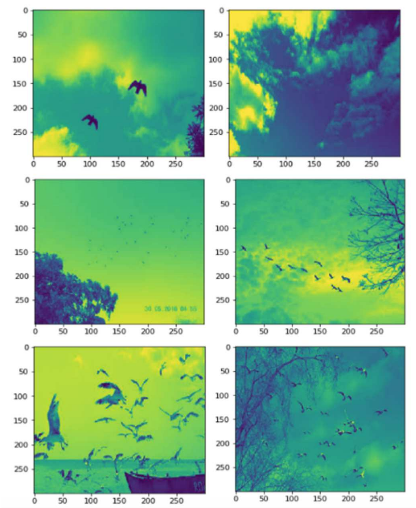

Fig. 4: Background images after pre-processing step. These pictures are specifically picked to trigger false positives with birds and some just show a blank background. These are then paired with the accompanying background noise before being used by our model.

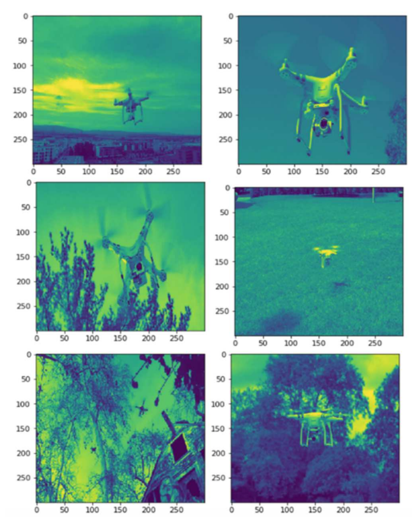

Fig. 5: Drone images after pre-processing step. Representative pictures from the dataset that show a drone in the frame. There are several types of drones in the dataset, but most have a very similar design to the ones above.

extremely small due to the filtering processing done for the image and RF data as discussed above. The size proved to be an issue since the methods used required numerous training examples. However, because the final dataset consists of cases where images are not optimal for detection (due to occlusion and visibility issues), the RF data can help. Nonetheless, model architectures and tuning can be performed with this simulated dataset. While the extremely small size would be a high risk for overfitting, we avoid this issue because we use real-time cameras, which generate huge numbers of images per minute, so our dataset ends up being much bigger, with us having more than 5000 thousand images, and using 1500 images.

V. Proposed Methodology

We propose a system capable of using both the image and RF data to make a drone detection decision. The proposed system is a feed-forward, artificial neural network with a deep and wide architecture. The input for the neural network is the high and low bands of the 2.4GHz frequency and an image. The output is a probability of the existence of a drone. The system currently provides a binary classification of the existence of a drone. Fig. 6 details the architecture of the proposed system. The object detection includes several tasks such as edge detections, semantic segmentation, localization, and other operations, which are preprocessing steps for any classification projects. While our techniques and overall methodology may seem complex, we used the typical deep learning computations. Additionally, we faced little difficulties in running our method using available computation resources. The innovation of our proposed approach enables us to incorporate sensor fusion, an already existing technique in a way that is cohesive. Additionally, because our proposed approach uses both high and low bands of frequency, another innovation is that the image quality is not jeopardized. Sometimes when working with images in drone detection, the quality of the image is compromised, leading to less accurate results. Our innovation emphasizes simplicity and ease of use, which is crucial for drone detection so that results can be interpreted better, and scalable.

Something to consider is that while the object detection task should consist of both localization and classification, there is a distinction between the two. Localization in drone detection focuses on finding the physical location of the drone under a real or virtual coordinate system. Classification in regards to drone detection deals with the problem of detecting the presence of objects in the image from a given set of object classes, without any localization. Additionally, classification can be treated in a binary manner, where two labels are used: 1 for a drone and 0 for not a drone. In our case, we specifically focus on detecting whether a drone is there or not; So in this regard, the classification aspect is more applicable for our experiment rather than localization; This is also why we have used a lot of classification architectures as well, which are talked about in subsequent sections.

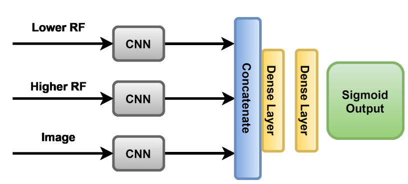

Fig. 6: Architecture of proposed system.

The CNNs and ANNs architectures used are customized for efficiency during the experimentation process. RNN results were not considered because they did not yield significant improvement over ANN. Additionally, [2] showed that ANNs can be used to successfully and accurately detect drones. A deep, wide ANN is used because the data from the RF sensor and the image are of different dimensions and semantics. As such, using the feature extractors allow for the final DNN to output a prediction based on both sensors. It also aided in the design process due to its inherently modular nature. The architecture was implemented module by module, doing some testing at each level. Initially, the subsystem of ANNs to detect drones just using the RF data was implemented and tested. The architecture for this subsystem is shown in Fig. 7.

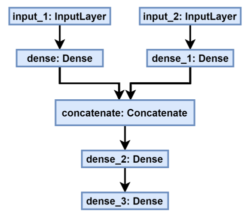

Fig. 7: ANN for classification using RF signals demonstrating two input networks for the upper and lower band of the RF data. The ouput is concatenated to a DNN which performs the prediction.

The ANN uses the ReLU activation function for the deep layers while the sigmoid function is used for the output layers to output a probability. After the implementation and testing phase of the ANN, CNN was implemented to recognize images. Successive convolutional, max pooling, and batch normalization layers were added during the testing and hyper-parameter tuning phase to achieve the best performance on the image dataset. The final model combined these two models to create the proposed system for drone detection. The final model is quite large, with $245,018,089$ trainable parameters that required high-performance computing to train. While the prediction is almost instant with a GPU, the size of the model hinders it from real-time use on the CPU. This does not include any data collection or data preprocessing time.

VI. Experiments and Results

A. Results for Initial ANN for RF-based classifications

Initially, each model used the raw RF dataset before the resampling procedure was implemented. The resulting models had to be small, with very few nodes per layer for the ANN and few convolutions for the CNN. Due to the size of the data, these approaches did not render any results and we ran into implementation issues where the required computation was too large. However, a small DNN was implemented following the architecture in Fig. 7 which yielded some acceptable performance. Table I details the performance metrics for this model. In Table I, class 0 is background and 1 is drone. We used a threshold of 0.5 and trained the model for 3 epochs. The metrics used were Precision, Recall, F1-Score, and Accuracy. Judging by the results in Table I, we noticed the precision was the same for both the background and drone classes. Overall, the performance was higher for the background class in comparison to the drone class, with accuracy being just 67%.

TABLE I: Performance metrics for initial ANN.

Performance metrics for initial ANN				
	Precision	**Recall**	**F1-Score**	**Support**
Background (0)	0.67	0.86	0.75	14
Drone (1)	0.67	0.40	0.50	10
Accuracy			0.67	24
Macro Avg			0.62	24
Micro Avg			0.65	24

B. Final Architecture

To improve results, two architecture changes were proposed and implemented. The first was a CNN with one-dimensional convolutional layers for RF-based identification. Regarding the CNN, we used a Leaky ReLU activation function, with the Adam optimizer. There were initial issues with the memory in implementation, but we managed to resolve these issues. Yet, even with some of these issues resolved, the CNN produces subpar performance when compared to the ANN. The second change was to use an RNN for RF-based identification as shown in Fig. 8. We can obtain this type of drone detection system by taking advantage of multiple components, and the number of frames that are captured. Because CNN is already designed to handle images, it makes it easier to integrate other elements in the architecture, thus guaranteeing optimal results.

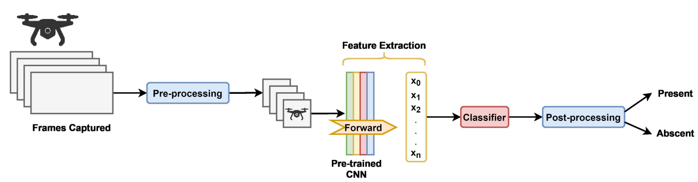

Fig. 8: Our proposed sensor fusion-based drone detection architecture.

C. The Final Model and Results

The final model consisted of the best performing parts of the initial model, which were further tuned after multiple training phases. However, the model exhibited some overfitting issues which could be either due to the limitations of the training data or constraints of the architecture itself. To mitigate the overfitting issues, We employed early stopping criteria. This resulted in a validation accuracy of 79% at the highest and 58% at the lowest in our training/testing phases. To assess the performance of the model better, we computed precision and recall at multiple thresholds and then determined the average precision-recall score from the resulting curve. As such, Fig. 9 presents the precision-recall curve of our final model.

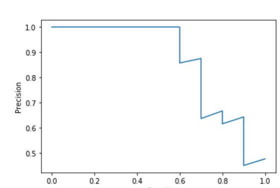

Fig. 9: Precision-recall curve for the final model.

Looking at Fig. 9, the average-precision recall score was 87%, which is rather high. The model performed with an average precision-recall score of about 78% on other trials. However, these results were not consistent and varied greatly with different training and testing sets. To combat this issue and get a better understanding of the model performance, we used K-fold cross-validation as shown in Table II. This is attributable to the small size of the dataset which made the validation set extremely small. Examining Table II and Fig. 9, the maximum recall can reach 1 because of how the CNN and ANN components are used for classification. The IoU metric in this experiment is evaluating how close an annotation or test output lines up with the ground truth. In this case, the IoU metric evaluates how well the model can differentiate between a drone and something that is not a drone. K-fold cross-validation can be used for optimization for preventing overfitting, but it can also be used as an evaluation tool for determining how well models perform, especially when the data is limited. As such, K-fold cross-validation is a suitable tool to use for our experiment.

TABLE II: Final model experimental results.

Metrics	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Avg.
Precision	0.80	0.70	0.83	0.86	0.78	0.79
Recall	0.50	0.88	1.00	0.55	1.00	0.78
F1-Score	0.62	0.78	0.91	0.67	0.88	0.77
Sensitivity	0.50	0.88	1.0	0.55	1.0	0.78
Specificity	0.88	0.63	0.91	0.80	0.78	0.80
Mathoew's Correlation Coefficient	0.40	0.52	0.87	0.32	0.78	0.58
Threshold	0.50	0.50	0.50	0.50	0.50	0.50
Training Accuracy	0.9531	0.9844	0.9375	0.6250	0.8750	0.7750
Training Loss	0.1496	0.1493	0.1694	0.1043	0.2206	0.1586
Validation Accuracy	0.6875	0.7500	0.9375	0.6250	0.8750	0.7750
Validation Loss	0.5618	0.4021	0.3410	0.5136	0.3643	0.4366

Regarding Table II, K-fold cross-validation was performed due to the small size of the data, and we set the value of k to 5 folds. It should be noted that we did use training, validation, and testing for our data; We split the data into 70% training, 20% testing, and 10% validation. Two epochs were chosen as a form of early stopping as overfitting became an issue. Additionally, Fig. 10 presents the results of a confusion matrix for one testing set. Proposing our sensor fusion architecture led to the best results we had received yet. We were able to run using still images and videos and receiving remarkable results. It performed spectacularly on photos containing other aircraft in the sky as shown in Fig. 11, where it can identify the drone apart from the helicopter with high confidence. Something to consider in our experiment is that we were only focusing on the class of the drone appearing in the situation, in other words, just focusing on detecting the actual drone. While it would have been interesting to have results for background situations, we considered this out of scope for our experiment.

VII. Conclusions and Future Work

The use of an additional sensor allows for better detection of drones, but it comes with some challenges. Analyzing the data from two different sources requires few preprocessing steps and a large ANN architecture to extract features and make a prediction. We implemented a deep and wide architecture

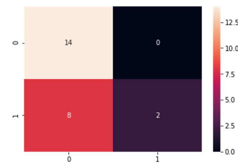

Fig. 10: Confusion matrix for one testing set.

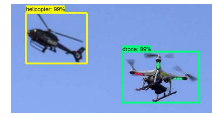

Fig. 11: One of our real-time experiments that identified the drone apart from the helicopter with high confidence.

ANN with a CNN to process the input image, two ANNs for the RF data, and a final DNN which produced an output probability. The large nature of this model and pre-processing steps required for the data raises some issues with the real-time use of such a system. However, with a GPU, the model can perform predictions in real-time. While our model successfully identified drones on most occasions, the model performance was great in comparison to current systems like [2] using a single sensor. The dataset that was used was also created with images and RF recorded separately. Overall, despite the challenges of fusing RF and image datasets, the metrics in Table II prove the feasibility of our study especially in cases of false positives like birds when compared with current implementations.

References

[1] C. Aker and S. Kalkan. Using deep networks for drone detection. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2017.

[2] M. F. Al-Sa'd, A. Al-Ali, A. Mohamed, T. Khattab, and A. Erbad. Rf-based drone detection and identification using deep learning approaches: An initiative towards a large open source drone database. *Future Generation Computer Systems*, 100:86–97, 2019.

[3] M. S. Allahham, M. F. Al-Sa'd, A. Al-Ali, A. Mohamed, T. Khattab, and A. Erbad. Dronerf dataset: A dataset of drones for rf-based detection, classification and identification. *Data in brief*, 26:104313, 2019.

[4] J. J. De Wit, R. I. Harmanny, and P. Molchanov. Radar micro-doppler feature extraction using the singular value decomposition. In *2014 International Radar Conference*, pages 1–6. IEEE, 2014.

[5] S. Jovanoska, M. Brötje, and W. Koch. Multisensor data fusion for uav detection and tracking. In *2018 19th International Radar Symposium (IRS)*, pages 1–10. IEEE, 2018.

[6] S. Samaras, E. Diamantidou, D. Ataloglou, N. Sakellariou, A. Vafeiadis, V. Magoulianitis, A. Lalas, A. Dimou, D. Zarpalas, K. Votis, et al. Deep learning on multi sensor data for counter uav applications—a systematic review. *Sensors*, 19(22):4837, 2019.

[7] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[8] F. Svanstrom, C. Englund, and F. Alonso-Fernandez. Real-time drone detection and tracking with visible, thermal and acoustic sensors. *arXiv preprint arXiv:2007.07396*, 2020.

[9] B. Taha and A. Shoufan. Machine learning-based drone detection and classification: State-of-the-art in research. *IEEE Access*, 7:138669–138682, 2019.

[10] H. Tan. If you're surprised that atlanta has the busiest airport on earth, you're not alone.