


Article

Low-Cost Raspberry-Pi-Based UAS Detection and Classification System Using Machine Learning

Carolyn J. Swinney ^{1,2,*}  and John C. Woods ¹

¹ Computer Science and Electronic Engineering Department, University of Essex, Colchester CO4 3SQ, UK; woodjt@essex.ac.uk

² Air and Space Warfare Centre, Royal Air Force, Lincoln LN5 9NB, UK

* Correspondence: cjswin@essex.ac.uk

Abstract: Small Unmanned Aerial Systems (UAS) usage is undoubtedly increasing at a significant rate. However, alongside this expansion is a growing concern that dependable low-cost counter measures do not exist. To mitigate a threat in a restricted airspace, it must first be known that a threat is present. With airport disruption from malicious UASs occurring regularly, low-cost methods for early warning are essential. This paper considers a low-cost early warning system for UAS detection and classification consisting of a BladeRF software-defined radio (SDR), wideband antenna and a Raspberry Pi 4 producing an edge node with a cost of under USD 540. The experiments showed that the Raspberry Pi using TensorFlow is capable of running a CNN feature extractor and machine learning classifier as part of an early warning system for UASs. Inference times ranged from 15 to 28 s for two-class UAS detection and 18 to 28 s for UAS type classification, suggesting that for systems that require timely results the Raspberry Pi would be better suited to act as a repeater of the raw SDR data, enabling the processing to be carried out on a higher powered central control unit. However, an early warning system would likely fuse multiple sensors. These experiments showed the RF machine learning classifier capable of running on a low-cost Raspberry Pi 4, which produced overall accuracy for a two-class detection system at 100% and 90.9% for UAS type classification on the UASs tested. The contribution of this research is a starting point for the consideration of low-cost early warning systems for UAS classification using machine learning, an SDR and Raspberry Pi.

Keywords: unmanned aerial vehicles; unmanned aerial systems; interference; UAS detection; RF spectrum analysis; machine learning classification; deep learning; convolutional neural network; transfer learning; signal analysis



Citation: Swinney, C.J.; Woods, J.C. Low-Cost Raspberry-Pi-Based UAS Detection and Classification System Using Machine Learning. *Aerospace* **2022**, *9*, 738. <https://doi.org/10.3390/aerospace9120738>

Academic Editors: Keumjin Lee and Hailong Huang

Received: 28 July 2022

Accepted: 18 November 2022

Published: 22 November 2022

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



Copyright: Crown Copyright © 2022. This material is licensed under the Open Government License v3.0 except where otherwise stated. To view this license, visit (<http://nationalarchives.gov.uk/doc/open-government-licence/version/3/>).

1. Introduction

A 2022 report by the Allied Market Research forecast stated that the small drone industry, which generated over USD 7 billion dollars in 2020, would reach USD 24 billion in 2030 [1]. It is undeniable that the market continues to grow as the economy reaps the benefits of the use of small UASs. Their wide range of uses vary from public service activities such as search and rescue to the advantages they provide for logistical movements and security functions. It is not just the commercial sector that is utilising UASs. Recently, United States President Biden gifted “100 Tactical Unmanned Aerial Systems” to support Ukraine in the conflict against Russia [2]. However, it seems the increased use of small UASs is not being matched by an equal interest in counter-UAS systems, which are required to combat malicious small UASs. This was recently highlighted in an article published by the Royal United Services Institute, stating how the way we fight is being shaped by small UASs and questioning whether military organisations are ready to face them [3]. Arguably, the largest incident of UAS disruption happened in 2018 at Gatwick Airport where over 1000 flights were grounded in the lead up to Christmas at a cost of over 50 million pounds to the UK economy [4]. It has been three and a half years since that occurrence but incidents

are still causing disruption all over the world. As this paper was written, in July 2022, the Ronald Reagan National Airport, Washington, D.C., United States, suspended flights for 45 min after a sighting of a small UAS. The United States Department of Homeland Security officials further revealed they received over 2000 sightings near airports in the last year [5]. Flights were diverted from the East Midlands Airport in the UK in June 22 due to the sighting of a small UAS [6]. In 2021, Marine Gen. Mckenzie, head of the United States Central Command, stated that concerns over small UASs were “amplified by the fact that dependable countermeasures against these drones currently don’t exist” [7]. Before a countermeasure can be employed, a system must be able to know that there is an unwanted UAS present in an airspace.

Both academia and industry have considered early warning systems from the perspective of fusing different sensors together. Zhang et al. [8] use a combination of sensors to improve detection capabilities. They use deep learning to perform object detection, LiDAR to work out the distance and thermal sensors to track the UAS. Shi et al. [9] use acoustic, imagery and RF sensors with machine learning classifier Support Vector Machine to perform detection. The fusion is performed using a logical OR function, and the system also performs localisation and counter measures by way of jamming. European projects Safeshore [10] and the Aladdin project [11] are two examples of industry-led programs. Safeshore uses data fusion for various detection methods. The Aladdin project incorporates detection, localisation, classification and neutralisation, using deep learning to fuse all the data together. If these systems work as discussed, then the question must be asked as to why incidents as described earlier are still happening; why are these systems not more widely utilised? One such answer could be cost. For this reason, our experiments have focused on the implementation of a low-cost early warning system, proving the utilisation of software-defined radio (SDR) with a Raspberry Pi to provide detection and classification of a UAS radio frequency (RF) signal. The use of RF and multiple SDRs in an early warning system provides future utility for localisation techniques.

Medina et al. [12] prove the use of a Raspberry Pi to process information received from a HackRF SDR. Khoi et al. [13] introduce object detection for small UAS using a Raspberry Pi processor and show promising results. Ozkan et al. [14] consider various types of deep learning models for the detection of UAVs using a Raspberry Pi as the platform. This work is purely focused on object detection using imagery. In the context of jamming, signal detection and classification research does exist, which utilises an SDR and a Raspberry Pi for real-time classification. Price et al. [15] detect and classify jamming signals, which could be used to interrupt a 2.4 GHz control signal between a UAS and ground control station. Price et al. produce jamming types barrage (Gaussian noise), protocol aware, single tone (cosine) and pulse using GNU-Radio and train a random forest classifier model. A small UAS is then fitted with a low-cost SDR, the HackRF One and a Raspberry Pi so that the classifier can make predictions on jamming signals in real time. They achieve a detection rate of 93% and suggest further work to include testing the system while the UAS is in operation and using the result to trigger mitigations (e.g., path re-scheduling). RF boasts advantages over object detection in terms of both detection range and the ability to perform other functions such as triangulation if multiple SDRs are used as edge devices. Various academic research exists with respect to utilising an SDR to perform RF-based detection and classification of UASs. Nei et al. [16] use a USRP-X310 SDR to perform classification along with research conducted by [17,18]. A USRP-B210 SDR is used in [19], USRP 2943R RF in [20] and a Lime SDR in [21]. Swinney and Woods use the BladeRF to perform classification in [22,23]. In these papers, the SDR is connected to a PC or laptop to perform the classification. In this paper, we extend this previous research to consider an early warning system in which an edge device would be a Raspberry Pi connected to an SDR—a low-cost system that could be widely used to alert potential airspace disruption.

Section 2 details the methodology including model creation and the implementation of the early warning system. Section 3 includes the results from both the model training

and validation and the early warning system. Lastly, Section 4 finishes with the conclusions and suggestions for future work.

2. Materials and Methods

2.1. Detection and Classification Model Creation

The training dataset used was based on the DroneDetect Dataset: A Radio Frequency dataset of Unmanned Aerial System (UAS) Signals for Machine Learning Detection and Classification [24]. Table 1 [25] shows the UAS considered in these experiments and the datalinks used for their transmission. For these experiments, the platforms were restricted where possible to the 2.4 GHz frequency range due to the use of one SDR. Future work is to include a second SDR operating in the 5.8 GHz range to cover both operating frequency ranges.

Table 1. UAS Transmission Systems.

Platform	Datalink	EIRP (2.4 GHz)	Freq Range (2.4 GHz)
Mavic 2 Air S	OcuSync 3.0	20 dBm	2.400–2.4835 GHz
Parrot Disco	Wi-Fi	19 dBm	2.400–2.4835 GHz
Inspire 2 Pro	Lightbridge 2.0	17 dBm	2.400–2.483 GHz
Mavic Pro 2	OcuSync 2.0	20 dBm	2.400–2.4835 GHz
Mavic Mini	Wi-Fi	19 dBm	2.400–2.4835 GHz

Each recording in the DroneDetect dataset consists of 1.2×10^8 complex samples equating to 2 s recording time in the form of a “.dat file”. In the experiments, the recordings were split into samples equating to 80 ms in length. The real and imaginary parts of the signal were added together and the samples processed in python using the Matplotlib API to produce spectrograms and power spectral density (PSD) graphs. The graphs were saved as images of 224×224 pixels to produce datasets of 250 samples per class, where 200 were used to train the system and 50 to validate the results with k-fold cross validation. PSD and spectrogram graphs were plotted using a 1024 FFT and a Hanning window with a 120 overlap. Figures 1 and 2 show a spectrogram and PSD, respectively, with no UAS present. It can be seen in Figure 2 that the noise floors sits around -77 dBm, and in Figure 1, some light background noise can be observed by the small, faint, yellow specs on the spectrogram. Figures 3 and 4 show a spectrogram and PSD, respectively, with a DJI Inspire flying. The platforms flew at a height of 20 m in a 40 m radius around the antenna with the pilot and controller approximately 4 m from the detection/classification system. In both plots, a wider concentrated band of larger bursts of activity can be observed in the higher end of the frequency band and also higher-powered, smaller bursts of activity, shown in a stronger yellow on the spectrogram, across a wider part of the spectrum.

Figures 5 and 6 show a spectrogram and PSD, respectively, with a DJI Mavic Mini. If Figures 3 and 5 are compared, there is a clear difference visually between the transmission of the Mavic Mini and the Inspire. The same is observed when comparing the PSD graphs in Figures 4 and 6.

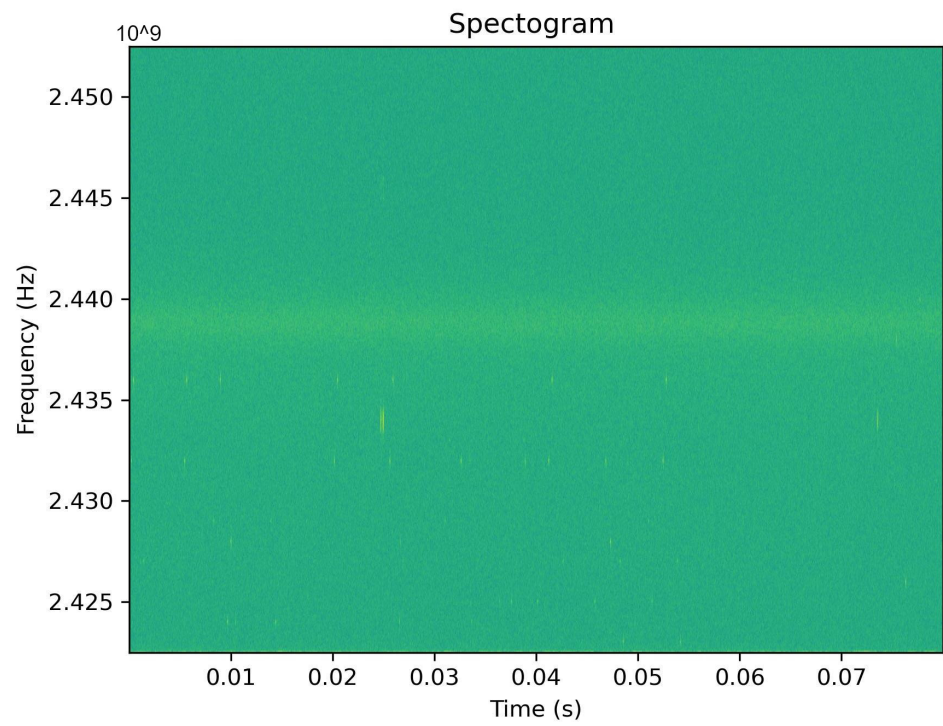


Figure 1. Spectrogram: no UAS present.

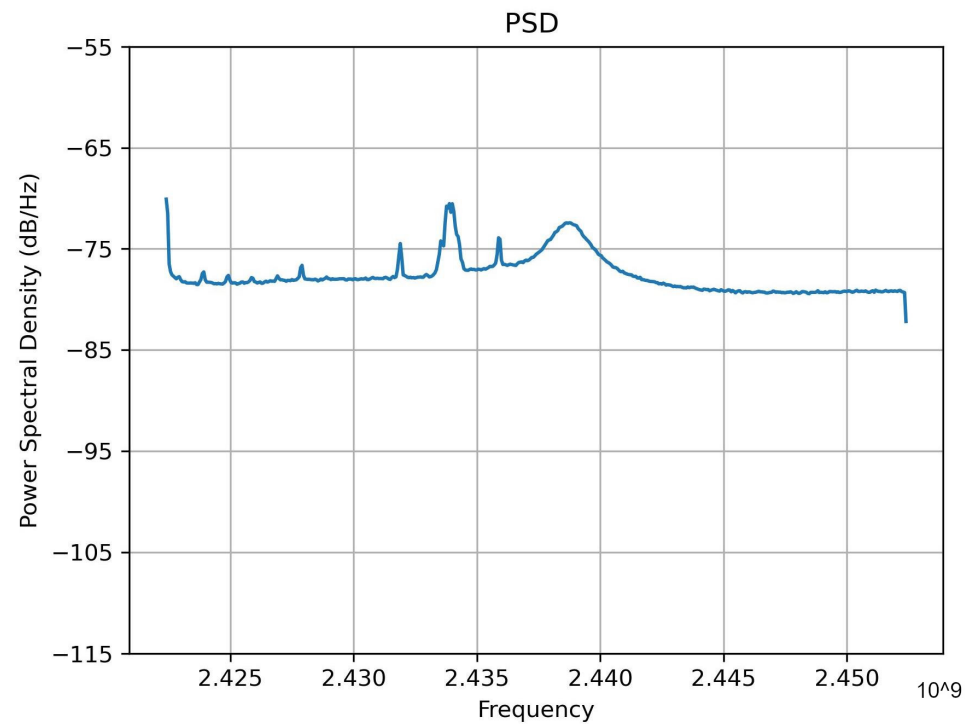


Figure 2. PSD: no UAS present.

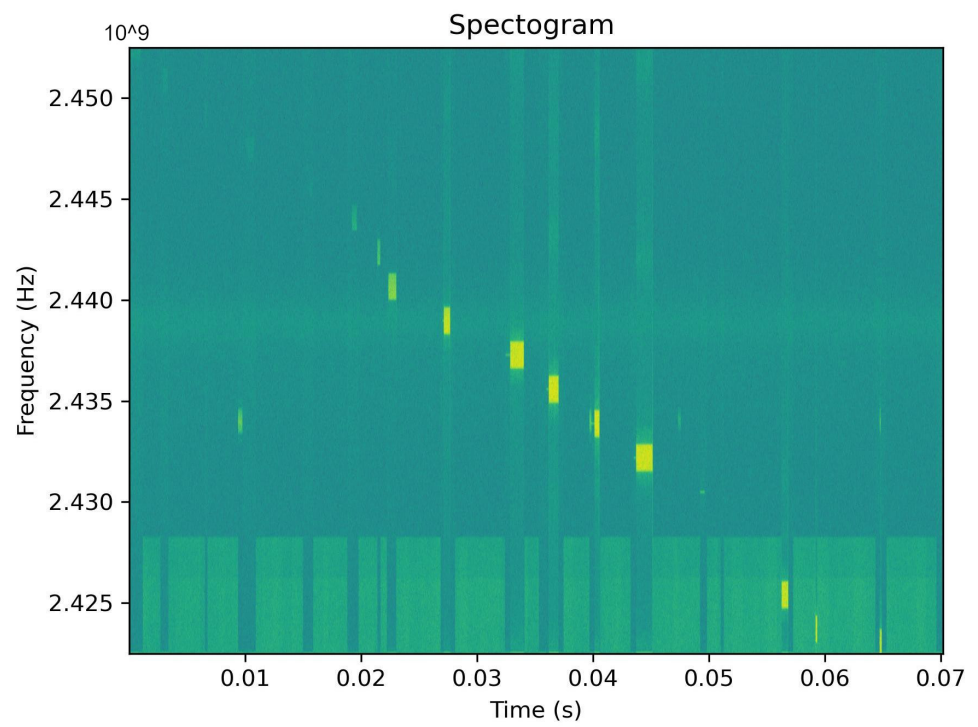


Figure 3. Spectrogram: DJI Inspire 2.

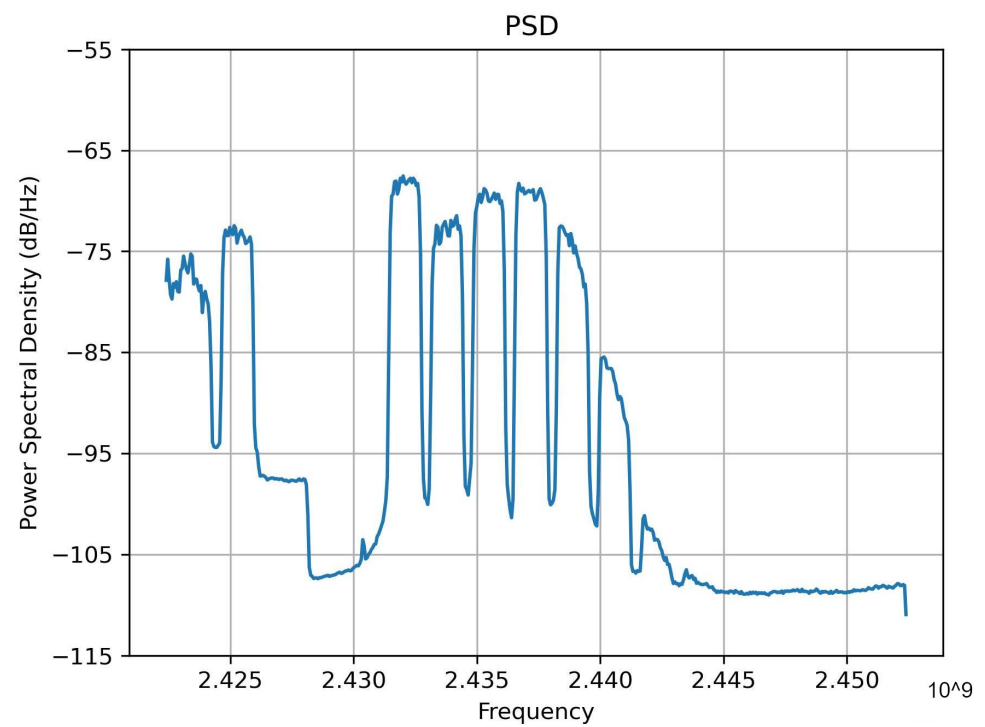


Figure 4. PSD: DJI Inspire.

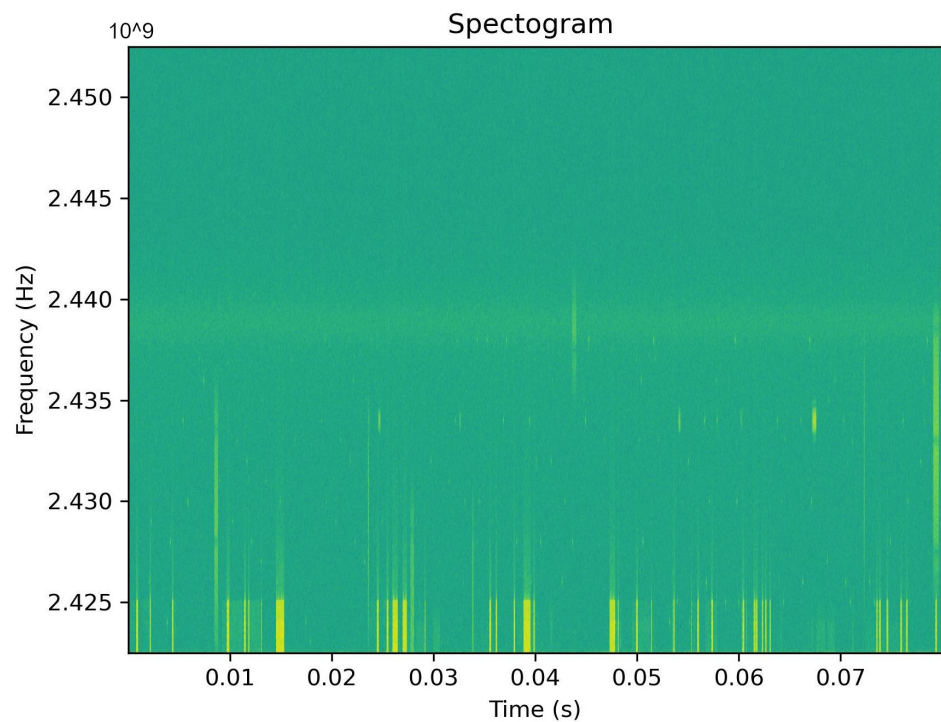


Figure 5. Spectrogram: DJI Mavic Mini.

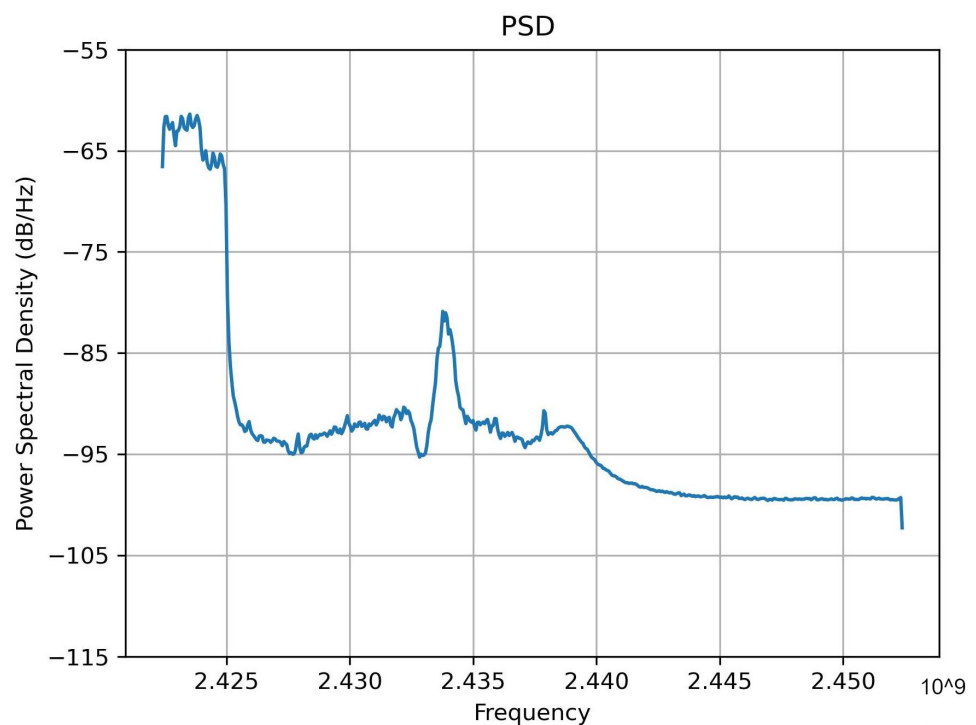


Figure 6. PSD: DJI Mavic Mini.

Using the training datasets for spectrograms and PSD images alongside a VGG-16 CNN with pre-trained weights on ImageNet, an object detection database of over 14 million images [26] was used as a feature extractor. During the training process, forward propagation was stopped at the last pooling layer and produced extracted features. The features were then used by machine learning classifiers logistic regression (LR) and k nearest neighbour (kNN) to produce the classification model. The output from the VGG-16

CNN is a feature vector of 25,088 values. Machine learning models and then uses this feature vector as an input to train the model. Models were produced for spectrogram and PSD images, for classifiers LR and kNN and for 2-class detection and 6-class UAS type classification. Five-fold cross validation was used to try and highlight any overfitting, and hyperparameters were optimised using 3-fold nested cross validation for regularisation and the number of neighbours for kNN. Models were saved using the python pickle module. LR and kNN were chosen as the machine learning classifiers to cover a linear and non-linear classifier. Linear classifiers such as LR have been shown to work well following CNN FE [27] and are quick to train. kNN was chosen to also show a non-linear classifier and understand whether there were any significant performance differences between the two.

2.2. Early Warning Implementation

For these experiments, the processing of the data to perform the classification was conducted on a low-cost Raspberry Pi acting as an edge device. The Raspberry Pi can be purchased for USD 35 [28], the BladeRF SDR by Nuand at USD 480 [29] and the Palm Tree Vivaldi Antenna at USD 18.99 [30], making the cost for one edge device to be under USD 540. Figure 7 shows the configuration of an early warning system with 3 edge devices made up of an antenna, SDR and Raspberry Pi, and one control unit. Using more than one edge device allows for extended coverage, for example, to cover the airfield scenario and perform other RF-enabled functions such as triangulation. Although the set up that has been described here could be used in a standalone manner, as the experiments have shown, it really depends on the requirements of the early warning system as to how you would employ the equipment detailed here. The scenario detailed in Figure 7 is that of an airfield, where ATC represents the air traffic control tower, a central location for housing a control unit that is consistently manned. The edge devices can then be placed around the airfield creating coverage across sensitive areas where aircraft are most vulnerable to small UAS interference.

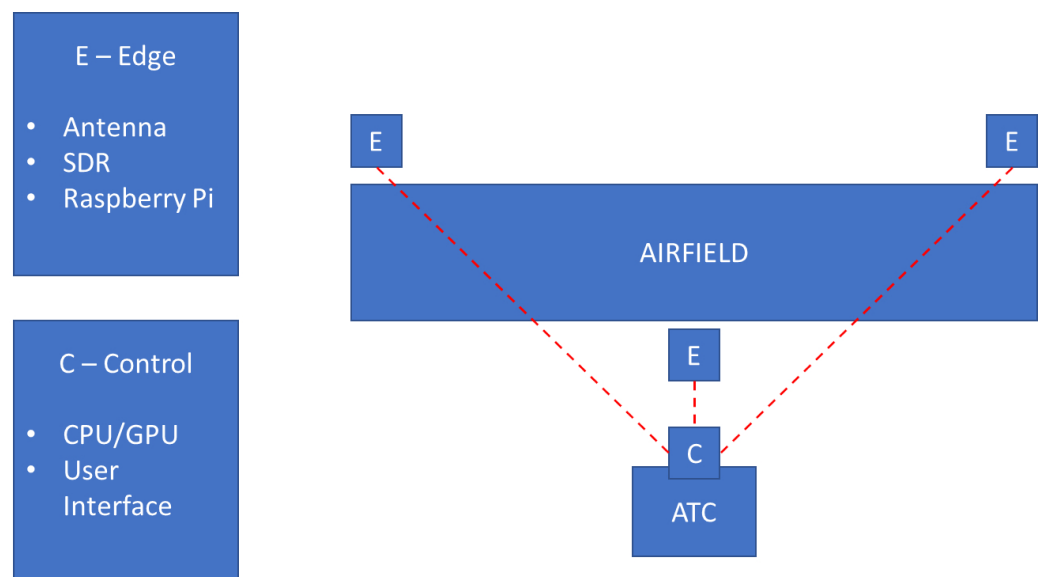


Figure 7. Early warning system configuration.

The control unit could simply be a laptop if the processing occurs on the edge devices. However, a higher powered processor could be required if the edge devices are used only to send the RF data back to the control unit to process there. For this reason, we also considered ZeroMQ sockets to transmit the data between the edge and control units.

GNURadio was used to read the data from the BladeRF SDR and send it out through a ZeroMQ socket. ZeroMQ is an open source messaging and communications library that is asynchronous and fast. ZeroMQ has different types of sockets depending on the type of

communication required, for example, request and reply is used when a reply is required for each message sent. In these experiments, publish and subscribe is used. This is where a publisher can send data and multiple recipients can subscribe to receive it. This method was chosen as an early warning system and may have two c2 nodes for redundancy, which both process the data. Another methodology would be to perform the processing at the receiving node; however, this would require more computational power at the end units. There are advantages and disadvantages of both approaches but ZeroMQ is capable of supporting either implementation with minor programming changes. ZeroMQ also supports pipelines for connected nodes and pairs for an exclusive connection.

Figure 8 shows the GNURadio set up using a ZeroMQ socket, which publishes the BladeRF data. GNURadio can be run on a microcomputer such as a Raspberry Pi, which when connected to an SDR provides a small footprint for an edge node in an early warning system.

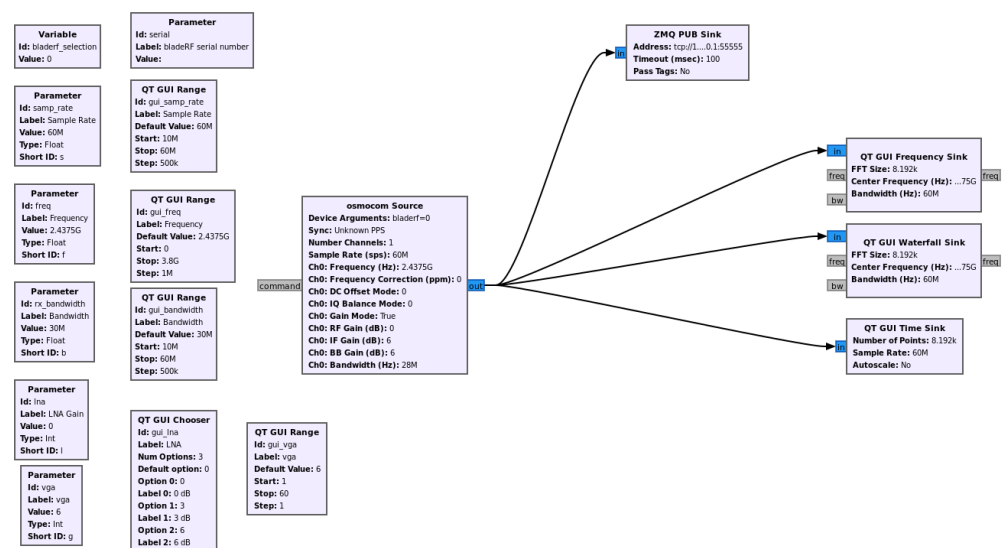


Figure 8. GNURadio set up with ZeroMQ socket.

On the c2 node, a python script would then receive the information from the socket to produce a graphical representation of the signal as an image and run it through the prediction model. Within the experiments, the socket information is received on the Raspberry Pi so that the time taken to make a prediction is evaluated using a low-cost edge device. The inference time is recorded from when the libraries are loaded and the information is being received from the socket until the prediction is made. Another consideration is that the control unit would likely be taking inputs from other sensors. For example, an early warning system may also include another sensor such as a video system or radar, which when activated would instigate the RF model to be run. The system could also include an unsupervised algorithm, which may have less accuracy but would produce another indication marker in a very quick time scale.

When we have the RF data and have produced the graphical signal representations, TensorFlow lite was used on the Raspberry Pi to load the previously trained model and make a prediction on the class. TensorFlow lite is a small version of the TensorFlow library specifically designed to run on Linux-based embedded devices, such as the Raspberry Pi. The Raspberry Pi in the experiments was loaded with Ubuntu 22.04, running python version 3.10.4 and TensorFlow version 2.9.0. Figure 9 shows a picture of the Raspberry-Pi-based UAS early warning system running the experiments on Ubuntu with python and TensorFlow. Figure 9 shows a picture of the Raspberry-Pi-based UAS early warning system running the experiments on Ubuntu with python and TensorFlow. Both Figures 9 and 10 show the live experimental set up for which the results can be seen in Section 3.2

(Early Warning Results), whereby the full system is validated with a Mavic Mini and a Mavic Inspire.

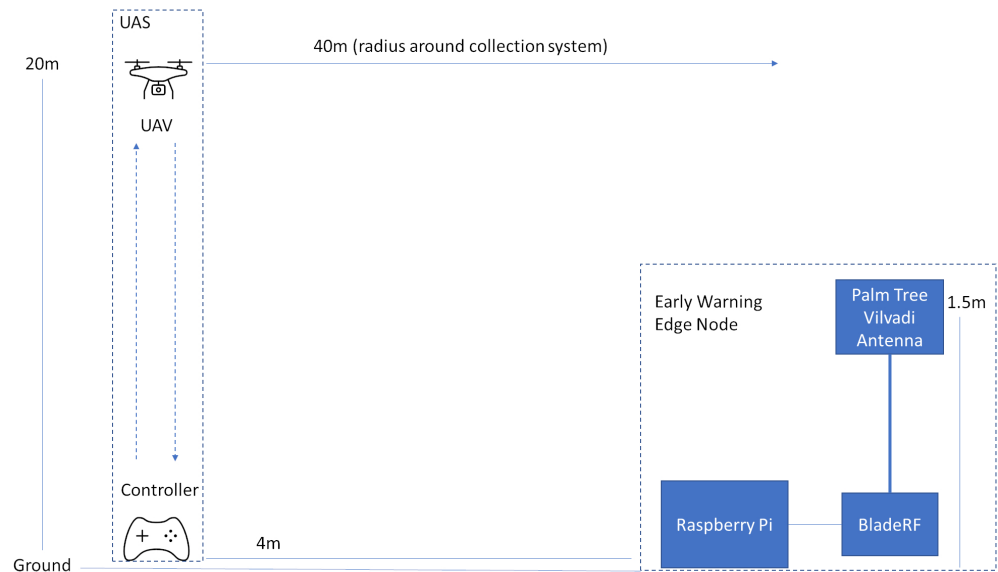


Figure 9. Raspberry-Pi-based UAS early warning system.

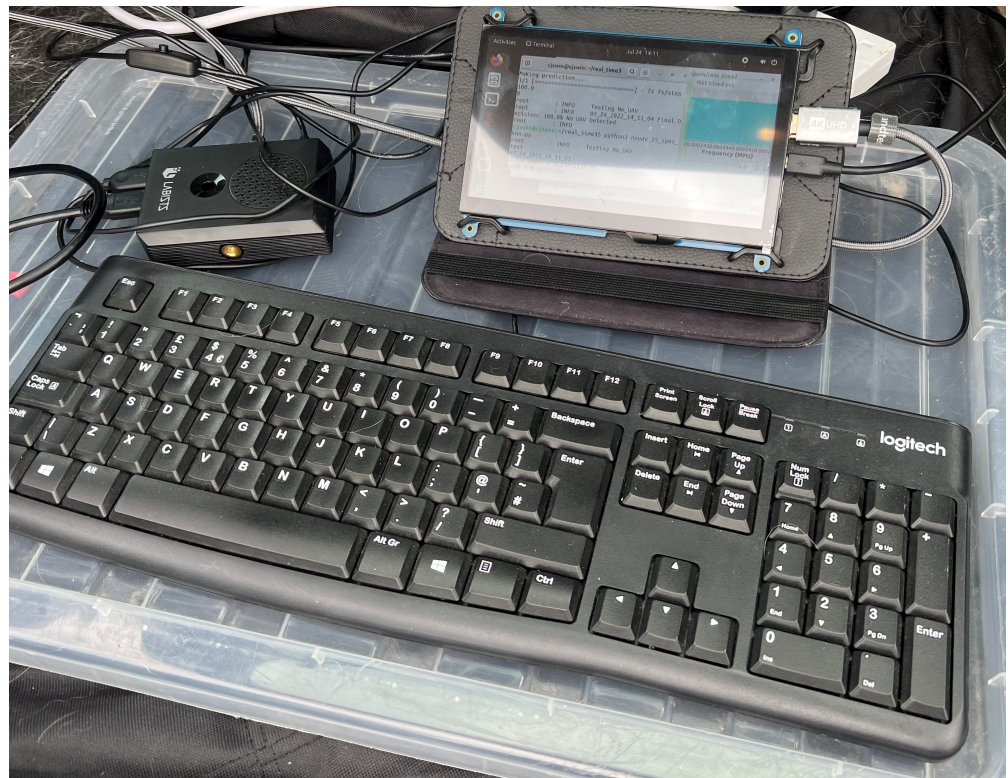


Figure 10. Early warning system setup.

Figure 10 shows the set up of the experiments to test the early warning system. The system captures any RF signals within a 28 MHz bandwidth with a 2.4375 GHz centre frequency. The BladeRF SDR was set with a sample rate of 60 MBits/s and connected to a low-cost antenna with a frequency range from 800 MHz to 6 GHz [31,32]. GNURadio ran on the Raspberry Pi to visualise the spectrum and also to show the implementation of

the ZMQ socket. A python script then ran in the terminal to receive the data and run the prediction using the previously trained models.

3. Discussion

3.1. Model Training and Validation

Before the early warning system is considered, the results from training the model are evaluated using F1-score and accuracy as the performance metrics. The metric accuracy considers how many times the model was right, while F1-score also takes into account the metrics of recall and precision. Recall calculates the fraction of positives predictions the model deems to be correct while precision considers the number of positive predictions that are in fact positive. Table 2 gives the F1-score and accuracy scores for the different models trained for two-class detection and six-class UAS type classification. It can be observed that PSD graphical signal representations slightly outperform spectrograms and that the LR models again slightly outperform kNN, but only marginally, in both cases. In the following tables, the performance metrics accuracy and F1-score are annotated as “acc” and “F1”, and spectrograms are annotated as spec.

Table 2. Results Accuracy (%) and F1-Score (%).

Classifier	Image	Metric	Detection	Type Classification
LR	PSD	Acc	100 (± 0.0)	99.3 (± 0.6)
	PSD	F1	100 (± 0.0)	99.2 (± 0.6)
	Spec	Acc	99.6 (± 0.3)	98.4 (± 0.6)
	Spec	F1	99.6 (± 0.3)	98.4 (± 0.6)
kNN	PSD	Acc	100.0 (± 0.0)	97.0 (± 0.6)
	PSD	F1	100.0 (± 0.0)	97.0 (± 0.6)
	Spec	Acc	98.2 (± 0.5)	95.7 (± 1.5)
	Spec	F1	98.2 (± 2.6)	95.6 (± 1.5)

To ensure the models were not overfitting, some data were held back for validation. Table 3 shows the validation results. When comparing Table 2 with Table 3, it can be seen that the models do not appear to be overfitting as the validation results do not drop significantly when the model is presented with new information.

Table 3. Validation Results Accuracy (%) and F1-Score (%).

Classifier	Image	Metric	Detection	Type Classification
LR	PSD	Acc	100	100
	PSD	F1	100	100
	Spec	Acc	98.6	98.5
	Spec	F1	98.6	98.5
kNN	PSD	Acc	99.3	97.7
	PSD	F1	99.3	97.7
	Spec	Acc	93.3	92.9
	Spec	F1	93.4	92.9

For logistic regression, hyperparameter optimisation values for regularisation [100, 10, 1.0, 0.1, 0.01] were tested using three-fold nested cross validation. The optimum value for regularisation for both spectrograms and PSD was 100.

Figure 11 below shows the confusion matrix for the kNN classifier with PSD graphical signal representations. We can observe that the majority of the misclassification occurs between the Ocusync and Lightbridge platforms (the Lightbridge being an predecessor to

the Ocusync). The largest misclassification is between the Air 2 S, Inspire 2 and the Mavic Pro 2. We also see some misclassification with the DX8. However, the misclassification is small.

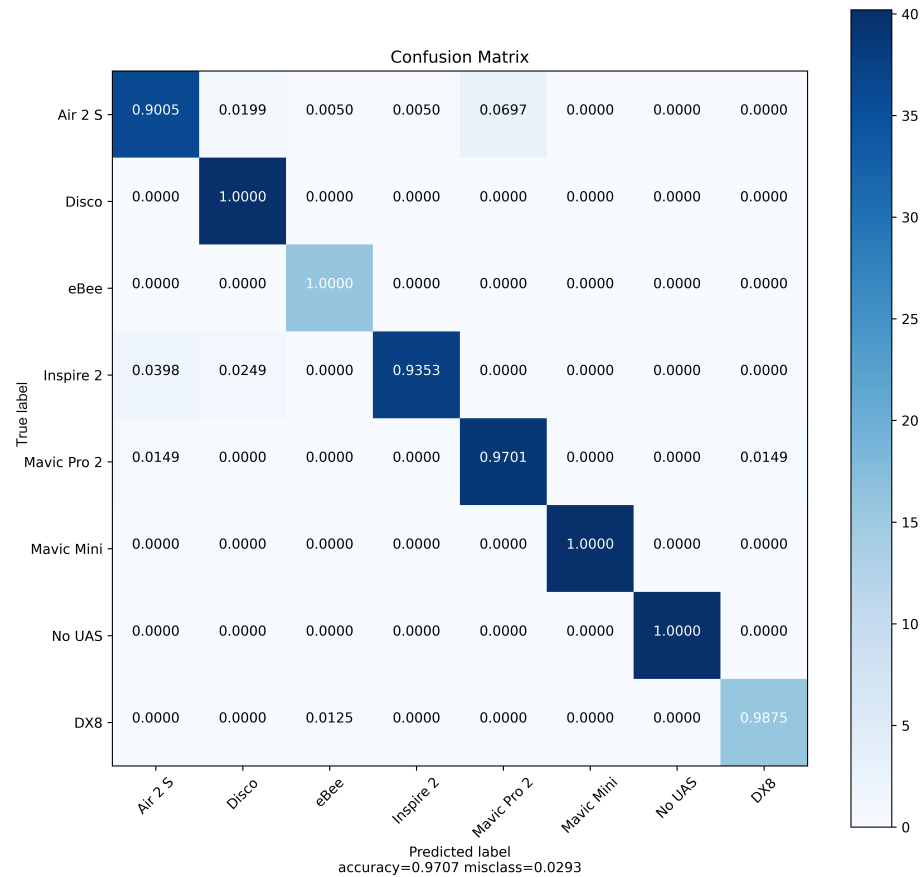


Figure 11. Confusion matrix PSD kNN.

For comparison, Figure 12 below shows the confusion matrix for the LR classifier with PSD graphical signal representations. Comparing Figure 11 with Figure 12, it can be observed that while LR still has a slight misclassification between the Ocusync/Lightbridge datalinks, it is classifying the DX8 correctly. This decreases the overall misclassification to under 1%.

Figure 13 shows the confusion matrix for LR using spectrogram images. A wider spread of misclassification occurs when using spectrograms including those between the three Ocusync/Lightbridge platforms and also the eBee and no UAS classes. However, this still produces an overall higher-performing classifier in terms of accuracy than the kNN model in Figure 11.

3.2. Early Warning Results

These models were then loaded onto the Raspberry Pi to be tested as part of the early warning system. Table 4 shows the results from running the two-class detection system on the Raspberry Pi. It can be observed that running each model in the presence of no UAS flying, Mavic Mini and the Mavic Inspire produced the correct prediction results with 100% confidence each time. Inference time varied from 15 to 28 s, lending itself to the conclusion that edge processing on a Raspberry Pi should either be used in conjunction with other sensors, which can produce a more timely result, or the Raspberry Pi should act as a relay with the processing being performed on a higher-powered device on the control unit. Overall, the two-class detection system was correct with its prediction on whether a UAS was present or not with 100% accuracy and 100% confidence.

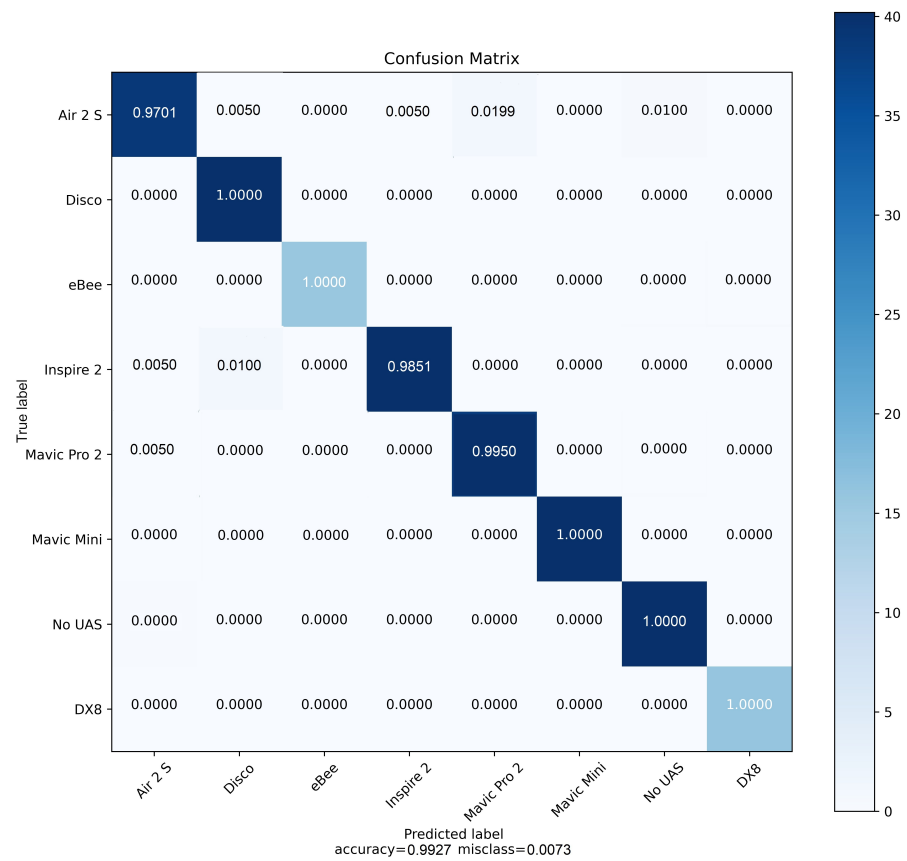


Figure 12. Confusion matrix PSD LR.

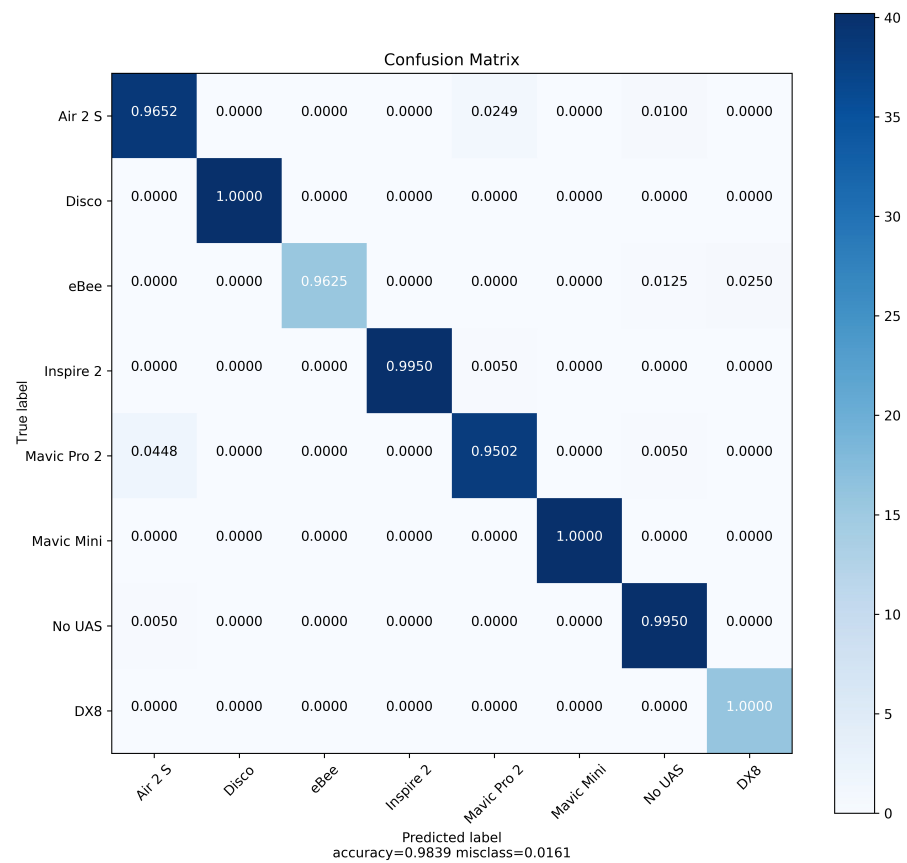


Figure 13. Confusion matrix spectrogram LR.

Table 4. Early warning two-class detection results.

Classifier	Image	UAS Flying	Model Prediction	Prediction (%)	Time (s)
LR	PSD	No UAS	No UAS	100	28
	PSD	Mini	UAS Detected	100	26
	PSD	Inspire	UAS Detected	100	15
	Spec	No UAS	No UAS	100	22
	Spec	Mini	UAS Detected	100	23
	Spec	Inspire	UAS Detected	100	19
kNN	PSD	No UAS	No UAS	100	24
	PSD	Mini	UAS Detected	100	24
	PSD	Inspire	UAS Detected	100	20
	Spec	No UAS	No UAS	100	24
	Spec	Mini	UAS Detected	100	26
	Spec	Inspire	UAS Detected	100	20

Table 5 shows the results from running the six-class UAS type classification system on the Raspberry Pi. Comparing the inference times in Tables 4 and 5, it can be observed that there is no real significant difference or penalty for performing a greater number of classes. Two-class detection inference times range from 15 to 28 s and those for the six-class UAS type classification system from 18 to 28 s. In terms of prediction accuracy, the system is 100% correct and confident in its prediction for no UAS and for the Mavic Mini. For the Inspire, the prediction model predicts correctly two out of three times but confidence in the prediction ranges from 50 to 66.67%. Table 5 shows that the highest confidence results are seen using the kNN classifier with PSD graphical signal representations. The overall accuracy was 90.9% for UAS type classification on the UASs tested. Further research would be needed to look at the reason why the Inspire produced lower confidence results; the original dataset may not have included all of the Lightbridge 2.0 activity. It is unlikely that the loss of confidence was due to environmental changes, as the Mavic Mini and No UAS predictions were 100% correct with 100% confidence.

Table 5. Early warning six-class UAS type classification results.

Classifier	Image	UAS Flying	Model Prediction	Prediction (%)	Time (s)
LR	PSD	No UAS	No UAS	100	22
	PSD	Mini	Mini	100	27
	PSD	Inspire	Inspire	50	18
	Spec	No UAS	No UAS	100	22
	Spec	Mini	Mini	100	24
	Spec	Inspire	-	-	-
kNN	PSD	No UAS	No UAS	100	26
	PSD	Mini	Mini	100	27
	PSD	Inspire	Inspire	66.7	23
	Spec	No UAS	No UAS	100	27
	Spec	Mini	Mini	100	28
	Spec	Inspire	Air 2 S	60	21

4. Conclusions

The experiments showed that the Raspberry Pi 4 B connected to a BladeRF SDR and low-cost antenna is capable of running a CNN feature extractor and machine learning classifier as part of an early warning system for UASs. However, the inference times ranged from 15 to 28 s for two-class UAS detection and 18 to 28 s for classification. This suggests that for systems that require timely results, the Raspberry Pi would be better suited to act as a repeater of the raw SDR data. This would enable the production of the graphical signal representations and the machine learning model prediction to be completed on a

higher-powered central control unit. If time was not of a concern, then the Raspberry Pi would be capable of making predictions as an edge device and could easily form part of a larger system made up of other sensors capable of faster indications to trigger higher accuracy results such as these. The overall accuracy of the two-class detection system was 100% and 90.9% for UAS type classification on the UAS tested, noting that three of the predictions for the classifier ranged from 50 to 66.67% in confidence. This research provides a starting point for the consideration of low-cost early warning systems for UAS detection and classification using machine learning, an SDR and a Raspberry Pi. Previous work in the field has concentrated on object detection using imagery and proved successful. RF and SDRs as edge devices boast other advantages such as triangulation and longer detection ranges, which makes them a good option as part of future early warning systems.

Further research would be needed to understand the reason why the Inspire produced lower confidence results for six-class UAS type classification. It is possible that the original dataset may not have included all of the potential Lightbridge 2.0 activity, which may move around the frequency band dependent on other activity. It is unlikely that the loss of confidence was due to environmental changes as the Mavic Mini and No UAS predictions were 100% correct with 100% confidence. For these experiments, the platforms were restricted to the 2.4 GHz frequency range due to the use of one SDR. However, all of the platforms considered in these experiments are capable of auto switching between 2.4 GHz and 5.8 GHz. A future piece of work would include a second SDR operating in the 5.8 GHz range to cover both operating frequency ranges. Future testing should also include a wider range of sensors and consider how the data could be integrated and fused within a control unit, as described in Figure 7. This is a larger piece of work that could also consider tracking the UAS using the RF data from multiple SDRs to triangulate the signal. Future work should include the consideration of metrics such as false alarm rate, training and testing time, and memory consumption. These metrics would add further valuable information for comparison with higher-end systems, allowing researchers to consider the trade-off between the cost of an early warning system and its performance.

Author Contributions: Conceptualization, C.J.S. and J.C.W.; methodology, C.J.S. and J.C.W.; software, C.J.S.; validation, C.J.S. and J.C.W.; investigation, C.J.S.; resources, C.J.S. and J.C.W.; data curation, C.J.S.; writing—original draft preparation, C.J.S.; writing—review and editing, C.J.S. and J.C.W.; visualization, C.J.S. and J.C.W.; supervision, J.C.W.; project administration, C.J.S. and J.C.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data not yet publicly available.

Acknowledgments: This work was carried out through the support of the School of Computer Science and Electronic Engineering, University of Essex, UK and the Royal Air Force, UK. Special thank you to Pilot Jim Pullen for flying the UAVs needed to produce the dataset for this work.

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

References

1. Himanshu, J.; Mutreja, S. Small Drones Market Size, Share, Growth, Analysis, Trends 2030. *Allied Market Research*, February 2022. Available online: <https://www.alliedmarketresearch.com/small-drones-market> (accessed on 27 July 2022).
2. Dilanian, K.; De Luce, D.; Kube, C. Biden admin will provide Ukraine with killer drones called Switchblades. *NBC News*, 15 March 2022. Available online: <https://www.nbcnews.com/politics/national-security/ukraine-asks-biden-admin-armed-drones-jamming-gear-surface-air-missile-rcna20197> (accessed on 27 July 2022).
3. Cranny-Evans, S. As Small Drones Shape How We Fight, is the British Army Ready to Face Them? *RUSI*, 21 July 2022. Available online: <https://www.rusi.org/explore-our-research/publications/commentary/small-drones-shape-how-we-fight-british-army-ready-face-them> (accessed on 27 July 2022).
4. Shackle, S. The mystery of the Gatwick drone. *The Guardian*, 2020. Available online: <https://www.theguardian.com/uk-news/2020/dec/01/the-mystery-of-the-gatwick-drone> (accessed on 23 June 2021).
5. Gilmer, E.D.C. Airport Incident Exposes Gaps in Counter-Drone Authorities. *Bloomberg Government*, 22 July 2022. Available online: <https://about.bgov.com/news/d-c-airport-incident-exposes-gaps-in-counter-drone-authorities/> (accessed on 27 July 2022).

6. BBC News. Airport disruption after drone sightings near Download Festival. *BBC News*, 10 June 2022. Available online: <https://www.bbc.co.uk/news/uk-england-leicestershire-61763290/> (accessed on 27 July 2022).
7. McKenzie, K. US Army General: Small Drones Biggest Threat Since IEDs. *The Defense Post*, 10 February 2021. Available online: <https://www.thedefensepost.com/2021/02/10/small-drones-threat-us-general/> (accessed on 23 June 2021).
8. Zhang, Z.; Zeng, C.; Dhameliya, M.; Chowdhury, S.; Rai, R. Deep learning based multi-modal sensing for tracking and state extraction of small quadcopters. *arXiv* **2020**, arXiv:2012.04794v1.
9. Shi, X.; Yang, C.; Xie, W.; Liang, C.; Shi, Z.; Chen, J. Anti-Drone System with Multiple Surveillance Technologies: Architecture, Implementation, and Challenges. *IEEE Commun. Mag.* **2018**, *56*, 68–74. [\[CrossRef\]](#)
10. De Cubber, G.; Shalom, R.; Coluccia, A.; Borcan, O.; Chamrád, R.; Radulescu, T.; Izquierdo, E.; Gagov, Z. The SafeShore system for the detection of threat agents in a maritime border environment. *IARP Workshop on Risky Interventions and Environmental Surveillance*, 18–19 May 2017. [\[CrossRef\]](#)
11. Sherpa, P. Aladdin. Aladdin Project Sherpa Workshop. *Horizon 2020 European Union*, 12 April 2019, pp. 1–38. Available online: https://aladdin2020.eu/wp-content/uploads/2019/06/ALADDIN_SHERPA_WS_12April2019_VO1-1.pdf (accessed on 27 July 2022).
12. Medina, E.; Advisor, M.; Paradells, J. Drone Detection and Inhibition. Master’s Thesis, Universitat Politècnica de Catalunya, Barcelona, Spain, February 2020.
13. Khoi, T.Q.; Quang, N.A.; Hieu, N.K. Object detection for drones on Raspberry Pi potentials and challenges. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1109*, 012033. [\[CrossRef\]](#)
14. Ozkan, Z. Raspberry Pi Object Detection and Recognition of Unmanned Aerial Vehicles Using Raspberry Pi Platform. In Proceedings of the 2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) Ankara, Turkey, 21–23 October 2021; pp. 467–472.
15. Price, J.; Li, Y.; Shamaileh, K.A.; Niyaz, Q.; Kaabouch, N.; Devabhaktuni, V. Real-time Classification of Jamming Attacks against UAVs via on-board Software-defined Radio and Machine Learning-based Receiver Module. In Proceedings of the IEEE International Conference on Electro Information Technology, Mankato, MN, USA, 19–21 May 2022; pp. 252–256. [\[CrossRef\]](#)
16. Nie, W.; Han, Z.C.; Li, Y.; He, W.; Xie, L.B.; Yang, X.L.; Zhou, M. UAV Detection and Localization Based on Multi-Dimensional Signal Features. *IEEE Sens. J.* **2022**, *22*, 5150–5162. [\[CrossRef\]](#)
17. Basak, S.; Rajendran, S.; Pollin, S.; Scheers, B. Drone classification from RF fingerprints using deep residual nets. In Proceedings of the 2021 International Conference on COMMunication Systems and NETworkS, COMSNETS 2021, Bangalore, India, 5–9 January 2021; pp. 548–555. [\[CrossRef\]](#)
18. Ezuma, M.; Erden, F.; Anjinappa, C.K.; Ozdemir, O.; Guvenc, I. Detection and classification of UAVs using RF fingerprints in the presence of interference. *arXiv* **2019**, arXiv:1909.05429v1. [\[CrossRef\]](#)
19. Xu, C.; Chen, B.; Liu, Y.; He, F.; Song, H. RF Fingerprint Measurement for Detecting Multiple Amateur Drones Based on STFT and Feature Reduction. In Proceedings of the Integrated Communications, Navigation and Surveillance Conference, ICNS, Herndon, VA, USA, 8–10 September 2020. [\[CrossRef\]](#)
20. Nemer, I.; Sheltami, T.; Ahmad, I.; Yasar, A.U.H.; Abdeen, M.A. Rf-based UAV detection and identification using hierarchical learning approach. *Sensors* **2021**, *21*, 1947. [\[CrossRef\]](#) [\[PubMed\]](#)
21. Flak, P. Drone Detection Sensor with Continuous 2.4 GHz ISM Band Coverage Based on Cost-Effective SDR Platform. *IEEE Access* **2021**, *9*, 114574–114586. [\[CrossRef\]](#)
22. Swinney, C.J. RF Detection and Classification of Unmanned Aerial Vehicles in Environments with Wireless Interference. In Proceedings of the 2021 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 15–18 June 2021; pp. 1494–1498. [\[CrossRef\]](#)
23. Swinney, C.J.; Woods, J.C. The Effect of Real-World Interference on CNN Feature Extraction and Machine Learning Classification of Unmanned Aerial Systems. *Aerospace* **2021**, *8*, 179. [\[CrossRef\]](#)
24. Swinney, C.J.; Woods, J.C. DroneDetect Dataset: A Radio Frequency dataset of Unmanned Aerial System (UAS) Signals for Machine Learning Detection and Classification. *IEEE DataPort*, 2021. Available online: <https://iee-dataport.org/keywords/drone-detection> (accessed on 14 June 2021).
25. DJI. Inspire 2—Product Information— DJI. 2022. Available online: <https://www.dji.com/cn/inspire-2/info> (accessed on 27 July 2022).
26. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [\[CrossRef\]](#)
27. Swinney, C.J.; Woods, J.C. Unmanned Aerial Vehicle Operating Mode Classification Using Deep Residual Learning Feature Extraction. *Aerospace* **2021**, *8*, 79. [\[CrossRef\]](#)
28. Raspberry Pi. Buy a Raspberry Pi 4 Model B—Raspberry Pi. 2022. Available online: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/> (accessed on 27 July 2022).
29. Nuand. bladeRF 2.0. Available online: <https://www.nuand.com/bladerf-2-0-micro/> (accessed on 26 April 2021).
30. Tindie. Ultra-WideBand Vivaldi Antenna 800 MHz to 6 GHz+ from Hex and Flex. Tindie. Available online: <https://www.tindie.com/products/hexandflex/ultra-wideband-vivaldi-antenna-800mhz-to-6ghz/> (accessed on 23 June 2021).

31. Nassar, I.T.; Weller, T.M. A Novel Method for Improving Antipodal Vivaldi Antenna Performance. *IEEE Trans. Antennas Propag.* **2015**, *63*, 3321–3324. [[CrossRef](#)]
32. De Oliveira, A.M.; Perotoni, M.B.; Kofuji, S.T.; Justo, J.F. A palm tree Antipodal Vivaldi Antenna with exponential slot edge for improved radiation pattern. *IEEE Antennas Wirel. Propag. Lett.* **2015**, *14*, 1334–1337. [[CrossRef](#)]