

Gábor Farkas¹ 

Augmentation of Training Data for AI-based Drone Detection System

Abstract

The increasing use of FPV drones in modern conflicts necessitates compact and energy-efficient detection systems capable of operating in dynamic electromagnetic environments. AI-based RF signal detection is a promising solution. However, its application can be limited by the lack of labelled datasets and the constraints of embedded platforms. This article presents a method for generating and augmenting training data directly from signals captured from analogue FPV video transmitters. Finally, a convolutional neural network was trained using the generated dataset and evaluated in a real-time environment. Experimental results demonstrate reliable detection performance, indicating that the proposed method is an effective and efficient solution for embedded FPV drone detection systems.

Keywords: electronic warfare, machine learning, ESM, software-defined radio, CUAV, FPV drone

Introduction

The rapid proliferation of unmanned aerial systems (hereinafter UAS), and in particular first-person-view (hereinafter FPV) drones, has fundamentally altered the character of contemporary battlefield operations. Originally developed for civilian and recreational purposes, FPV drones have been rapidly adapted for military use due to their low cost, ease of modification, and high operational effectiveness.² Recent conflicts, especially the Russo–Ukrainian war, have demonstrated the large-scale use of FPV drones for reconnaissance, target acquisition, and direct kinetic attacks against personnel and equipment.³ Their widespread availability and expendability enable adversaries to deploy them in large numbers, creating a persistent and asymmetric

¹ PhD student, Ludovika University of Public Service, Doctoral School of Military Engineering, e-mail: farkas.gabor.csp@gmail.com

² SEO et al. 2023.

³ MOLLOY 2024: 17.

threat to ground forces. This transformation is consistent with the broader evolution of modern warfare and hybrid operational environments.⁴

Unlike larger UAS, FPV drones operate at low altitude, exhibit small radar cross-sections, and often rely on commercially available radio-frequency (hereinafter RF) components. As a result, traditional air defence and surveillance systems are frequently ineffective or economically impractical against such targets.⁵ From the perspective of troop protection, early detection of approaching FPV drones is therefore critical, as it directly increases reaction time and enables the application of countermeasures.⁶ Electronic support measures (ESM) as part of electronic warfare (EW), operating in the electromagnetic spectrum, provide a promising means of detecting these systems by exploiting their RF emissions rather than relying solely on optical sensors.⁷

FPV drones typically transmit a continuous video stream from an onboard video transmitter (hereinafter VTX) to the remote operator. These transmissions, most commonly located in the 1.2 GHz and 5.8 GHz bands, are characterised by relatively high output power and distinct spectral features. As the drone approaches its target, the strength of the received signal of the video link increases, making RF-based detection particularly suitable for short-range warning systems. Consequently, recognising FPV-related RF signals has become a key research topic within modern counter-UAS and EW applications.⁸

The long-term objective of this research is the development of a compact EW device, that can be carried by soldiers and capable of autonomously detecting approaching FPV drones and issuing an immediate warning. Such a system must operate in real time, be lightweight and energy-efficient, and function reliably in a highly dynamic and contested electromagnetic environment. The proposed concept is based on a software-defined radio (hereinafter SDR) front-end coupled with an embedded computing module, forming a mobile RF sensing platform.⁹ Within this architecture, artificial intelligence-based (hereinafter AI) signal recognition is employed to distinguish FPV drone transmissions from other RF activity present on the battlefield.¹⁰

The application of AI techniques to the classification of RF signals in embedded EW systems introduces several significant challenges. First, to ensure suitability for mobile, battery-powered operation, the neural network used must be as compact and computationally efficient as possible, while still maintaining adequate detection performance. Second, publicly available and labelled training datasets for FPV drone RF signals are essentially nonexistent. Even if such datasets were available, they would rapidly become outdated due to the evolution of drone hardware. Third, any practical detection system must be capable of incorporating newly captured RF signals and

⁴ HAIG et al. 2014.

⁵ HORVÁTH et al. 2024.

⁶ HELL 2017.

⁷ LEE et al. 2018.

⁸ POPPINGA et al. 2025.

⁹ FARKAS et al. 2025.

¹⁰ NÉMETH-VIRÁGH 2023.

generating updated training data to maintain effectiveness against emerging threats. Addressing these challenges requires methods that enable systematic creation and augmentation of training datasets directly from captured RF streams. This capability is essential for rapid adaptation and to keep pace with the fast-changing electromagnetic environment observed in modern conflicts. The focus of this article is therefore on determining and evaluating a possible method for generating and augmenting datasets suitable for AI-based FPV drone detection. The proposed approach leverages real-world RF captures obtained using an SDR-based EW framework and applies data augmentation techniques to compensate for limited training samples.

The contributions of this work are twofold. First, it presents practical methods for constructing and augmenting RF training datasets from raw SDR data, specifically tailored to FPV drone signal characteristics. Second, the resulting datasets are evaluated using an existing convolutional neural network (hereinafter CNN) architecture, which is subsequently integrated into a real-time RF capture environment. By validating the approach on embedded hardware under realistic conditions, this study aims to bridge the gap between laboratory-scale AI experiments and deployable EW systems capable of supporting troop protection on the battlefield.

Analogue FPV video stream

Detecting the VTX signal from FPV drones is particularly advantageous for RF-based warning systems. Compared to control links, the video downlink typically operates with higher transmit power, exhibits continuous transmission, and becomes progressively stronger as the drone approaches the receiver. Furthermore, analogue FPV video signals are highly repetitive and structured in time, making them well suited for pattern-based detection approaches.

Instead of fully demodulating or decoding the video content, which would impose significant computational overhead, the proposed method relies on a visualisation technique known as rasterisation. In this approach, consecutive samples of the captured RF stream are placed sequentially in image rows. Once a row is filled, subsequent samples continue in the next row. This process effectively converts a one-dimensional time-domain signal into a two-dimensional image representation. Although the semantic video content is not reconstructed, characteristic temporal structures remain clearly visible as seen in Figure 1.

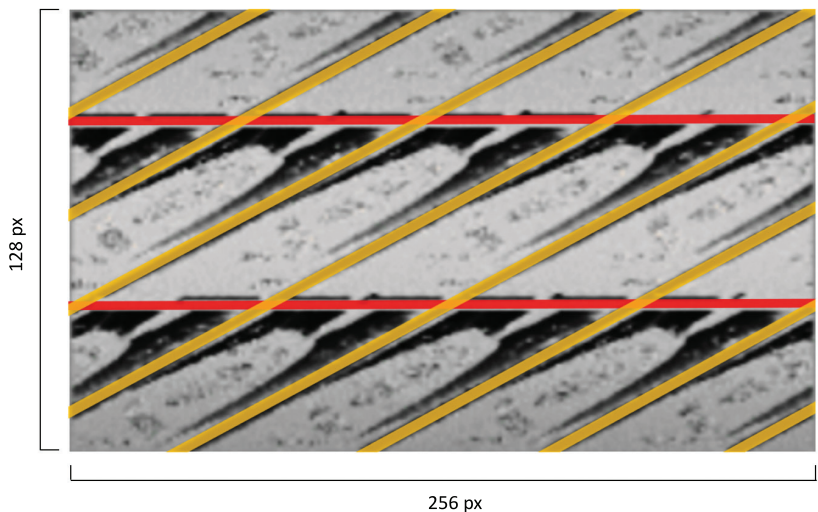


Figure 1: Synchronisation signals form a noticeable pattern on the rasterised image
Note: red line corresponds to vertical and yellow line to horizontal synchronisation
Source: compiled by the author

This representation closely resembles what a human observer would intuitively recognise as a video signal. Consequently, CNNs can be trained to identify these patterns automatically. Earlier work used an autocorrelation-based detection method, but its reliability proved insufficient under varying signal conditions and interference. The rasterised image-based approach could enable more robust detection by leveraging the spatial pattern recognition capabilities of CNNs.

Analogue FPV video standards

The majority of analogue FPV systems in current use rely on legacy composite video standards. While implementations may vary slightly across manufacturers, the fundamental timing characteristics remain consistent. The most encountered formats are summarised in Table 1.

Table 1: Typical analogue FPV video parameters

Standard	Frame rate	Horizontal line rate	Active lines	Approx. pixel clock
NTSC	29.97 Hz	15.734 kHz	480	12–14 MHz
PAL	25 Hz	15.625 kHz	576	14–15 MHz

Source: compiled by the author

These standards employ line-synchronous scanning, where each frame consists of a fixed number of horizontal lines, separated by synchronisation pulses.¹¹ For FPV detection purposes, precise pixel reconstruction is unnecessary. Only the repetitive timing structure of the signal is required.

Rasterised image construction

To generate a CNN-compatible input, the complex baseband IQ stream¹² captured by the SDR is first converted into a magnitude signal:

$$x[n] = \sqrt{I[n]^2 + Q[n]^2}$$

This operation removes phase information and produces a signal suitable for grayscale image generation. The magnitude samples are then rasterised into a two-dimensional array. In this work, an image resolution of 256×256 pixels is selected. This size provides a good compromise between computational efficiency and sufficient spatial resolution, and it closely matches the input tensor dimensions used by common CNN architectures such as ResNet-18.¹³

Sampling rate determination

To ensure that characteristic video patterns including horizontal and vertical synchronisation lines are preserved in the rasterised image, the SDR sampling rate must be carefully selected. To make the horizontal structure visually apparent, the image should contain multiple complete line crossings. In this work, the target is to display sixteen crossings in two rows to target a reliable detection. To visualise two vertical sync lines in the rasterised image we need to capture at least two frames. Therefore, the required time window is

$$T_V = \frac{2}{f_{frame}} = \frac{2}{30Hz} = 66.7ms$$

Taking into account the size of the image 256×256 pixels, the sampling rate can be determined in the following way:

$$f_{SR} = \frac{1}{\frac{T_V}{256 \times 256}} = \frac{1}{\frac{66.7ms}{65,536}} = 982.5kHz$$

¹¹ JACK 2011.

¹² IQ stream refers to a continuous sequence of sampled in-phase (I) and quadrature (Q) signal values acquired over time.

¹³ LIU et al. 2024.

This results in roughly two horizontal line intervals appearing within each image row block, which is sufficient to generate clearly distinguishable vertical and horizontal patterns in the rasterised image. Since many SDR platforms, like HackRF,¹⁴ allow only discrete sampling rates, a nominal value of 1 MSPS (millions of samples per second) is selected as a practical and widely supported configuration.¹⁵

Implications for dataset generation

With the sampling rate fixed and the rasterisation parameters defined, the SDR output can be consistently transformed into images that exhibit authentic FPV video patterns. These images form the basis for the training dataset used in subsequent CNN-based detection experiments. This makes it particularly suitable for real-time operation in dynamic battlefield RF environments.

Determining augmentation variables

When constructing a training dataset from captured FPV drone VTX signals, data augmentation is a necessity rather than an optional enhancement. Two primary motivations drive this requirement. First, RF signals encountered during operational deployment will inevitably differ from those recorded under laboratory conditions due to hardware variability, environmental effects and dynamic channel conditions. Second, in realistic field scenarios, only a limited amount of signal data may be available for capture. Consequently, the training dataset must be expanded synthetically to ensure sufficient variability and generalisation capability of the neural network.

Pixel frequency variability

Analogue FPV video transmitters are not frequency-stable precision instruments. Variations in component tolerances, oscillator drift, supply voltage and temperature can cause deviations in the effective pixel clock and line timing of the transmitted video signal. Practical measurements and manufacturer specifications indicate that these deviations typically fall within a range of approximately $\pm 1\%$ under normal operating conditions. In the rasterised image representation, such timing deviations manifest as a change in the slope or angle of the vertical synchronisation structures. A slightly higher effective pixel frequency causes vertical lines to tilt in one direction, while a lower frequency results in an opposite tilt. If the training dataset only contains perfectly aligned patterns, the CNN may overfit to a narrow representation and fail to generalise. To simulate this variability, the recorded RF stream is artificially under-sampled and over-sampled within a bounded range corresponding to the

¹⁴ HackRF One is an open source SDR hardware platform (<https://greatscottgadgets.com/hackrf/one/>).

¹⁵ HackRF documentation (s. a.).

expected pixel frequency deviation. This operation effectively introduces controlled geometric distortions in the rasterised images while preserving the underlying signal structure.

Signal-to-noise ratio variation

The clarity of the rasterised image is strongly influenced by the signal-to-noise ratio (hereinafter SNR) of the captured RF stream. In practice the SNR can vary significantly due to several factors, including distance between the drone and the detector, antenna orientation, multipath propagation, and the presence of other RF emitters operating in the same frequency band. To account for these effects, noise is added synthetically to the magnitude signal prior to rasterisation. This allows the generation of image samples corresponding to different SNR conditions. The range of applied Gaussian noise values is chosen such that the characteristic horizontal and vertical synchronisation patterns remain visually discernible. If the noise level is too high, the repetitive structure of the video signal is completely obscured, rendering the sample unusable for both human and machine recognition. Therefore, augmentation is constrained to a range where degradation is realistic but still informative for training.

Temporal offset of the rasterisation window

Each rasterised image is constructed from a fixed-length segment of the RF stream, consisting of 65,536 consecutive samples. In a continuous video transmission, the starting point of this segment is arbitrary. As a result, the captured window may begin in any phase relative to the video line or frame synchronisation. This offset causes horizontal and vertical features to appear at different spatial positions in the rasterised image. If the dataset only contains images with aligned structures, CNN may implicitly learn absolute positional dependencies rather than the presence of the pattern itself. To mitigate this effect, the starting index of the sampled segment varies during dataset generation. By randomly shifting the capture window within the continuous stream, the resulting images exhibit the same structural elements at different locations, promoting translation invariance in the trained model.

Scene-dependent video content

Although the proposed method does not decode the video signal, the analogue video content still influences the rasterised representation. Changes in brightness, contrast, and scene dynamics introduce variations in the amplitude envelope of the signal, which in turn affect the grayscale texture of the generated images. During laboratory

recordings, it is therefore advantageous to expose the FPV camera to scenes that are approximately expected operational environments, such as open terrain, vegetation, or mixed backgrounds. This ensures that the captured signal exhibits realistic amplitude fluctuations. In contrast, battlefield captures are inherently uncontrolled, and the observed video content may vary widely. Because it is not feasible to control video content in operational scenarios, the augmentation strategy must account for this uncertainty. By combining multiple laboratory recordings with different visual scenes and applying the previously described augmentation methods, the resulting dataset captures a broader range of signal appearances, improving robustness under real-world conditions.

Capturing RF signal for dataset generation

To generate training data suitable for AI-based FPV drone detection, a live analogue VTX signal must first be captured. Based on the previous analysis, the SDR sampling rate and rasterised image dimensions are fixed at 1 MSPS and 256×256 pixels, respectively. In this experiment, 1 second of data was captured that contains FPV VTX signal.

Frequency variation via resampling

As discussed previously, the effective pixel frequency of analogue FPV video signals may vary due to hardware tolerances and environmental effects. To simulate this variability, the recorded RF stream is resampled to emulate small deviations in timing. In this study, 100 resampled versions of the captured stream are generated, spanning a frequency variation range from -1% to $+1\%$ around the nominal sampling rate. For this experiment the resampling is performed in 100 steps. Each resampled stream represents a slightly different effective pixel clock and produces a distinct geometric appearance in the rasterised images.

Capture length

Each rasterised image is constructed from a fixed-length segment of the magnitude signal containing 65,536 samples. To enable sufficient variability in temporal alignment, multiple image windows are extracted from each captured and resampled stream using different starting offsets. In the proposed approach, the stream is divided into 40 partially overlapping segments, each producing one rasterised image.

Noise augmentation and image count

To model varying channel conditions and detection ranges, additive noise is applied to each image segment. Ten noise levels are used, ranging from 0% to 50% relative distortion, corresponding to progressively decreasing SNR while preserving recognisable structural patterns. Taking into account the 40 segments taken from the original stream, resampling applied in 100 steps and adding relative distortion gives the total number of generated training images all together 40,000. This dataset size is considered sufficient as an initial training set for a CNN architecture such as ResNet-18.¹⁶

Laboratory capture setup

The RF data used for dataset generation were collected under laboratory conditions using GNU Radio¹⁷ as the signal acquisition and visualisation environment. A HackRF One SDR was employed as the RF front-end. A simple GNU Radio flowgraph was constructed with the following functional blocks:

- Frequency scanning and tuning, enabling manual identification of active VTX carriers.
- Time Raster Sink, providing real-time visualisation of the incoming signal in a rasterised format. This visualisation closely resembles the images later generated for CNN input, enabling intuitive confirmation of signal presence and structure.
- File Sink, used to store raw complex baseband samples for offline processing and dataset generation.

The user interface for the flowgraph can be seen in Figure 2. This setup allows efficient discovery of VTX signals even when the exact carrier frequency is initially unknown. Once a suitable signal is identified, a short recording of a few seconds is sufficient to generate the full augmented dataset described above. As a signal source, a small "Tiny Whoop" class FPV drone¹⁸ equipped with an analogue video transmitter was used. Several seconds of continuous video transmission were captured at close range, resulting in a high-quality reference recording. Although the capture environment was controlled, the subsequent augmentation steps ensure that the generated dataset reflects a wide range of realistic operational conditions.

¹⁶ SHIN et al. 2016.

¹⁷ GNU Radio is an open-source software development toolkit for software-defined radio, enabling the construction of signal processing pipelines ("flowgraphs") using modular functional blocks (www.gnuradio.org/).

¹⁸ "Tiny Whoop" refers to a class of small FPV quadcopters with ducted propellers and low-power analogue video transmitters, typically used for short-range flight (www.tinywhoop.com/collections/bnf).

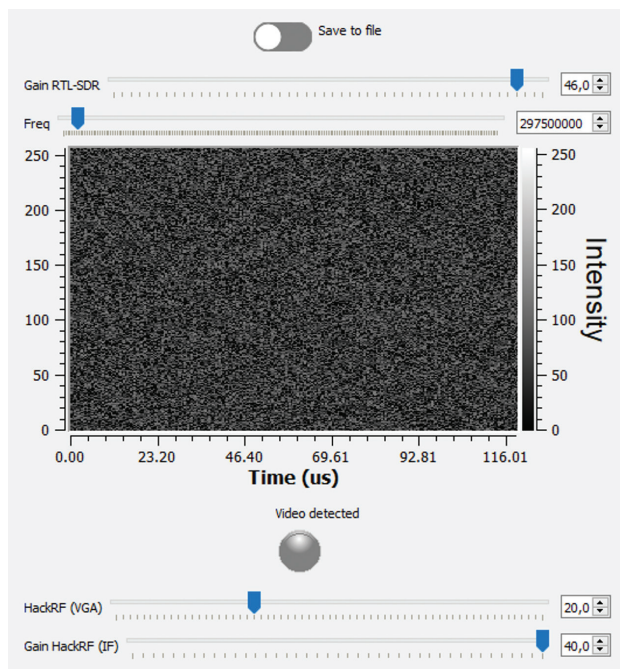


Figure 2: User interface of a GNU Radio project where the rasterised signal is displayed in real time

Note: additional controls can be seen for fast signal discovery

Source: compiled by the author

Implications for rapid dataset creation

The described capture and augmentation strategy enables the generation of a large and diverse training dataset from minimal raw RF data. This is particularly important in field scenarios, where capture opportunities may be limited to short time windows. By relying on systematic resampling, temporal offset variation and controlled noise injection, the approach supports rapid dataset preparation and iterative retraining, which are essential for maintaining detection performance in evolving electromagnetic environments.

Creating dataset

Once the RF capture is stored as a file containing complex IQ samples, the preparation of the training dataset can begin. All dataset generation and processing steps are implemented using Python scripts, chosen for their portability, readability and compatibility with common scientific computing environments. This design choice ensures that the dataset creation pipeline can be reproduced on a wide range of platforms, including embedded systems and standard development workstations.

Stream resampling and normalisation

As described previously, the first step in dataset preparation is the resampling of the initial portion of the captured RF stream. From the original recording, 100 resampled streams are generated to simulate small variations in effective pixel frequency. Each resampled stream is converted to a nominal sampling rate of 1 MSPS. The magnitude of each complex sample is computed and normalised to an 8-bit grayscale range. This normalisation ensures consistent intensity scaling across all samples and allows the resulting data to be directly represented as grayscale images without further conversion.

Manual validation of signal presence

Before proceeding with large-scale augmentation, a manual validation step is introduced to ensure data quality. Each resampled stream is split into 16 non-overlapping segments of 65,536 samples, corresponding to 16 rasterised images per stream. These images are saved to disk and visually inspected. This process yields a total of 1,600 images. The purpose of this step is to confirm that each resampled stream contains the expected FPV video patterns, such as stable horizontal and vertical synchronisation structures. Streams that do not meet these criteria due to interference, signal loss, or incorrect tuning are excluded from further processing. Introducing this validation stage prevents mislabelled or corrupted data from degrading training performance. Two rasterised images from the validation stage can be seen in Figure 3.

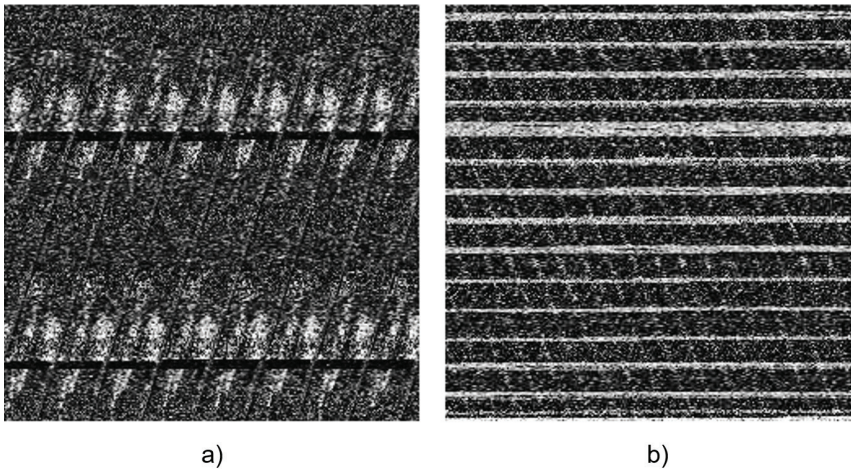


Figure 3: Captured RF stream converted to rasterised image

Note: a) signal from an FPV drone; b) signal from an unknown source

Source: compiled by the author

Randomised slice selection and SNR augmentation

For the remaining validated streams, the dataset is randomly distributed into three subsets before augmentation. This split follows common best practices in supervised learning and provides a balanced framework for model optimisation and unbiased performance evaluation:

- training set: 80%
- validation set: 10%
- test set: 10%

The dataset generation script selects random 65,536 sample slices from each stream to introduce temporal diversity. Noise is then applied to each slice at predefined levels to simulate varying SNR conditions. Each processed slice is rasterised into a 256×256 grayscale image and saved to disk. All images generated from FPV VTX recordings are labelled as *Signal*, representing the positive detection class that the neural network is intended to recognise.

Negative class (noise) dataset generation

To enable binary classification, an equally sized dataset representing the negative class is required. This class is labelled *Noise* and contains RF captures that do not include FPV video transmissions. The noise dataset is generated using the same processing pipeline as the signal dataset, ensuring consistent preprocessing and image characteristics. RF captures for this class include:

- background noise
- other communication signals (Wi-Fi, Bluetooth)
- unrelated RF activity within the monitored frequency bands

By maintaining symmetry in dataset size and processing steps, the risk of class imbalance and preprocessing bias is minimised. Once both classes are generated, the complete dataset consists of balanced *Signal* and *Noise* samples suitable for supervised CNN training. If additional FPV VTX recordings are later acquired, they can be incorporated by generating new *Signal* samples using the same procedure and merging them with the existing dataset. This modular approach supports incremental dataset expansion without requiring a complete regeneration of previously collected data. Representations of different amounts of noise applied to the stream can be seen in Figure 4.

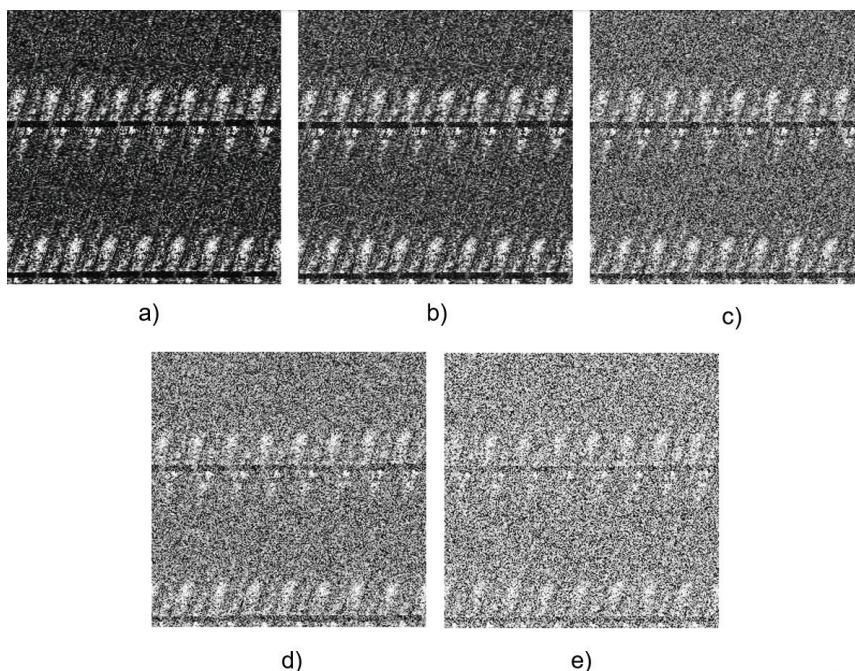


Figure 4: Different levels of noise applied to the same image

Note: a) 10%; b) 20%; c) 30%; d) 40%; e) 50%

Source: compiled by the author

Model training preparation

With the dataset prepared, the system is ready for neural network training. In this work, a ResNet-18 architecture is selected as the initial model due to its favourable trade-off between accuracy, computational complexity and suitability for embedded deployment. Training is performed using the PyTorch¹⁹ framework with transfer learning. The rasterised grayscale images are transformed into the appropriate input tensor format required by ResNet-18.²⁰ No additional image-domain augmentation techniques such as rotation, cropping, or flipping are applied, as the dataset already incorporates signal-domain variability through resampling, temporal shifting and SNR manipulation. Given the dataset size generated, training converges within a few minutes on standard computing hardware. This rapid training cycle is an important practical advantage, enabling frequent retraining and adaptation as new RF data becomes available.

¹⁹ PyTorch is an open-source machine learning framework used for building and training neural networks, particularly well-suited for deep learning applications (<https://pytorch.org/>).

²⁰ CHAPPELL-WHITEHEAD 2022.

Testing the model in real environment

After completing training, the performance of the neural network was first evaluated using the reserved test dataset. On this offline test batch, the model achieved a classification accuracy of 100%. While this result confirms correct dataset labelling and successful convergence, offline accuracy alone is insufficient to assess operational suitability. Therefore, further validation was conducted in a real-time RF acquisition and inference environment.

Model deployment and integration

To enable real-time inference, the trained ResNet-18 model was exported to the Open Neural Network Exchange (hereinafter ONNX) format. ONNX was selected due to its framework-independent representation and widespread support across inference engines and embedded platforms.

A custom Python processing block was implemented within the GNU Radio environment. This block operates directly on the live SDR data stream and performs the following steps:

- collection of 65,536 consecutive complex samples from the SDR output
- conversion of the IQ samples to magnitude values
- normalisation and reshaping of the data into a 256×256 grayscale image
- transformation of the image into the tensor format required by the neural network
- execution of model inference using the ONNX runtime

The model outputs class probabilities for the labels signal and noise. A detection event is declared if the confidence score for the signal class exceeds 90%. This threshold was chosen to reduce false alarms while maintaining high sensitivity to genuine FPV video transmissions.

Field testing

Real-time tests were conducted using three different analogue FPV video transmitters operating in a dense urban environment characterised by heavy RF background activity. The transmitters were mounted on small FPV platforms and moved within a radius of approximately 50 metres from the detector. To ensure a conservative evaluation, the VTX output power was limited to 100 mW. This power level is significantly lower than typical battlefield configurations, where FPV transmitters commonly operate at output powers exceeding 1 W. As a result, the test conditions represent a challenging detection scenario with reduced signal strength and increased interference. Under these conditions, the system achieved a detection success rate exceeding 96%.

False alarm assessment

To evaluate robustness against false detections, the system was operated continuously for an 8-hour period in the same urban RF environment, without any active FPV transmissions present. During this interval, the detector processed live RF data and performed inference continuously using the same confidence threshold. No false positive detections were recorded during the entire observation period. This result indicates strong discrimination between FPV video signals and background RF activity, including other communication systems operating in the monitored bands.

Discussion of results

The real-time experiments demonstrate that the proposed detection approach is not only effective under controlled offline conditions but also robust when deployed in a realistic and noisy RF environment. The combination of rasterised signal representation and CNN-based classification provides reliable detection even at reduced transmit power levels and in the presence of significant background interference.

While the observed performance metrics are highly promising, it should be noted that the tested scenarios represent short-range engagements and analogue FPV systems. Detection range and generalisation to other modulation schemes will depend on SDR sensitivity, antenna characteristics and further dataset expansion. Nevertheless, the absence of false alarms during extended operation is a particularly important result for practical deployment, where excessive false positives can significantly degrade operational usability.

Conclusion

This article addressed the problem of creating effective training data for AI-based detection of FPV drone video transmitters in scenarios where labelled RF datasets are unavailable, and signal characteristics change rapidly. The motivation arises from the operational need for compact, mobile electronic warfare systems capable of detecting approaching FPV drones in real time and under realistic battlefield conditions. The primary challenge is to enable reliable detection using limited captured data, while maintaining computational efficiency suitable for embedded, low-power platforms. To overcome this challenge, a complete workflow was presented, starting from live RF signal capture and extending through dataset generation, augmentation, neural network training and real-time deployment. The study demonstrated how a few seconds of captured analogue FPV video signal can be transformed into a sufficiently large and diverse training dataset through controlled resampling, temporal shifting and noise augmentation. The resulting dataset was used to train a convolutional neural network and validated both offline and in a real-time SDR-based detection environment.

A key contribution of this work is the use of image-based recognition applied to rasterised RF signals. Unlike common AI-based RF detection approaches that rely on frequency-domain representations such as FFTs or waterfall spectrograms, the proposed method operates entirely in the time domain. The preprocessing pipeline is computationally lightweight, requiring only magnitude calculation and rasterisation, which is significantly less demanding than spectral analysis. Compared to traditional autocorrelation-based detection, the CNN-based approach offers improved robustness and scalability, particularly when multiple repetitive signal types must be distinguished. The method is especially well suited for continuous and highly repetitive transmissions, such as analogue FPV video streams. It is emphasised that the presented approach represents only one possible solution among many for RF-based drone detection. Alternative signal representations, neural network architectures and detection strategies may be more appropriate for other signal types or operational constraints. Nevertheless, the results indicate that rasterised image-based recognition is a viable and effective technique for FPV drone detection in embedded EW applications. Future work will focus on designing and evaluating custom, lightweight CNN architectures tailored specifically to rasterised RF data. The objective is to reduce model size and energy consumption while preserving detection accuracy, enabling deployment on resource-constrained embedded platforms. Additional studies will also explore extending the approach to other signal classes and modulation schemes, further enhancing adaptability in dynamic electromagnetic environments.

References

- CHAPPELL, Andrew – WHITEHEAD, Leigh H (2022): Application of Transfer Learning to Neutrino Interaction Classification. *The European Physical Journal C*, 82(12). Online: <https://doi.org/10.1140/epjc/s10052-022-11066-6>
- FARKAS, Gábor – FAZEKAS, Gábor – NÉMETH, András (2025): FPV-drónok detektálásának alternatív megoldása konvolúciós neurális hálózattal. *Haditechnika*, 59(2), 2–7. Online: <https://doi.org/10.23713/HT.59.2.01>
- HackRF documentation (s. a.). Online: <https://hackrf.readthedocs.io/en/latest/index.html>
- HAIG, Zsolt – KOVÁCS, László – VÁNYA, László – VASS, Sándor (2014): *Elektronikai hadviselés*. Budapest: Nemzeti Közzolgálati Egyetem. Online: <https://opac.uni-nke.hu/webview?infile=&sobj=9276&source=webvd&cgimime=application%2Fpdf%0D%0A>
- HELL, Péter (2017): Drónelhárító rendszerek az objektumvédelemben. *Hadmérnök*, 12(3), 37–47. Online: www.hadmernok.hu/173_04_hell.pdf
- HORVÁTH, Tamás – ÖRDÖGH, Attila – BORKÓ, Máté (2024): Dróndetektáló fejlesztés. *Haditechnika*, 58(1), 55–59. Online: <http://doi.org/10.23713/HT.58.1.10>
- JACK, Keith (2011): *Video Demystified. A Handbook for the Digital Engineer*. Amsterdam: Elsevier.
- LEE, Dongkyu – LA, Woong Gyu – KIM, Hwangnam (2018): Drone Detection and Identification System Using Artificial Intelligence. *2018 International Conference on*

- Information and Communication Technology Convergence (ICTC)*, 1131–1133. Online: <https://doi.org/10.1109/ICTC.2018.8539442>
- LIU, Haixia – BRAILSFORD, Tim – BULL, Larry (2024): Resnet18 Performance: Impact of Network Depth and Image Resolution on Image Classification. *Proceedings of the 2024 8th International Conference on Advances in Artificial Intelligence*, 351–356. Online: <https://doi.org/10.1145/3704137.3704173>
- MOLLOY, Oleksandra (2024): *Drones in Modern Warfare: Lessons Learnt from the War in Ukraine*. Australian Army Research Centre. Online: <https://doi.org/10.61451/267513>
- NÉMETH, András – VIRÁGH, Krisztián (2023): Mesterséges intelligencia és haderő – Katonai alkalmazási lehetőségek VII. rész. *Haditechnika*, 57(1), 2–6. Online: <https://doi.org/10.23713/HT.57.1.01>
- POPPINGA, Gerald – ANDERSON, David – AMENDOLA, Danilo (2025): *Technical Developments in Counter-drone Technology: C-UAS Detection, Tracking and Identification Technology*. Luxembourg: Publications Office of the European Union. Online: <https://doi.org/10.2760/1517220>
- SEO, Kang-Il – CHO, Sang-Keun – PARK, Sang-Hyuk (2023): A Case Study on FPV Drone Combats of the Ukrainian Forces. *The Journal of the Convergence on Culture Technology*, 9(3), 263–270. Online: <https://doi.org/10.17703/JCCT.2023.9.3.263>
- SHIN, Hoo-Chang – ROTH, Holger R. – GAO, Mingchen – LU, Le – XU, Ziyue – NOGUES, Isabella (2016): Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE Transactions on Medical Imaging*, 35(5), 1285–1298. Online: <https://doi.org/10.1109/TMI.2016.2528162>