

MACHINE LEARNING IMAGE RECOGNITION FOR GNSS JAMMING SIGNALS CATEGORIZATION

*J. Steiner**, *J. Pešík†*

Abstract: Global Navigation Satellite Systems are a critical positioning, navigation, and timing source for various industries. However, their weak signal on Earth’s surface makes them vulnerable to jamming. This paper explores the use of machine learning image recognition for categorizing GNSS jamming signals. The study uses data from a long-term monitoring campaign, with over 2,000 jamming events recorded. Seven commonly used jamming signal types were analyzed using the Residual Neural Networks (ResNet). Five different ResNet models with 18 to 152 layers were evaluated, with the best performing achieving a precision greater than 90% in determining the correct jamming signal category.

Key words: *GNSS, machine learning, interference, jamming, image processing, classification*

Received: September 26, 2024

DOI: 10.14311/NNW.2024.34.019

Revised and accepted: December 31, 2024

1. Introduction

Global Navigation Satellite Systems (GNSS) have become an indispensable utility for today’s critical infrastructure, industry, and our everyday lives. As a source of positioning, navigation and timing (PNT), GNSS is an integral part of all transportation domains, the financial sector, data centres, and power grids to name a few examples [1].

Unfortunately, due to the nature of GNSS being a space-based PNT system, its signal is very weak on the Earth’s surface. The GNSS signal reaches a power level within the range of -165 to -150 dBW at the receiver antenna. This extremely low signal power makes the GNSS signal inherently vulnerable to radio frequency interference (RFI) [2, 3].

Intentional GNSS RFI in the form of jamming and spoofing has been on the rise for a few years now with jamming being a more frequent encounter. Jamming leads to a decrease in PNT information accuracy or a complete GNSS outage. GNSS jamming can be described as the transmission of a signal in the frequency

*Jakub Steiner; Czech Technical University in Prague, Faculty of Transportation Sciences, Konviktská 20, Prague 110 00, Czech Republic, E-mail: steinja8@fd.cvut.cz

†Jiří Pešík – Corresponding author; University of West Bohemia in Pilsen, Faculty of Economics, Univerzitní 22, Pilsen 30614, Czech Republic, E-mail: pesikj@fek.zcu.cz

bandwidth reserved for GNSS at a higher power than the authentic GNSS signal. GNSS jamming is especially common within and in the vicinity of war and conflict areas [4].

Although motivations for GNSS jamming vary outside of military usage, one of the main motivations includes avoiding paying the road toll. Modern tolling systems leverage GNSS to determine the distance a vehicle has travelled on a tolled road to calculate the final payment amount [5]. When GNSS is unavailable due to jamming, the vehicle may avoid paying the toll despite travelling the tolled road. Previous monitoring campaigns have demonstrated the frequency of GNSS jamming on tolled roads using GNSS jamming detectors. All monitoring campaigns show jamming to be a daily occurrence [6, 7, 8, 9].

It is important to note that individual jammers differ in the jamming signal they transmit. Unfortunately, jamming signal categories have not been standardised by any major authority yet. However, previous studies have suggested different frameworks based on the jamming signal characteristics [10]. The differences in individual jamming signals may be captured by the GNSS jamming detector. Some detector models provide a self-defined classification for jamming events or signal visualisation in the way of power spectral density, spectrum snapshots or waterfall diagrams. Some models even capture an I/Q signal snapshot for further analysis [11, 10].

GNSS jamming detectors provide valuable situational awareness and enable the activation of countermeasures against interference. However, these detectors primarily serve as reactive tools, identifying jamming events only after they occur. This limitation underscores the need for a proactive approach—one that can predict jamming events before they happen.

A proactive strategy relies on the premise that jamming events follow repetitive patterns linked to human activities, such as daily or weekly commuting routines. By conducting long-term jamming monitoring, patterns can emerge, revealing the likelihood of future jamming occurrences. If jamming events can be linked through their unique signal characteristics, predictive models can be developed to anticipate future incidents. This would enable more efficient law enforcement actions, improved GNSS service reliability, and targeted mitigation efforts.

To achieve this goal, this study explores the use of machine learning (ML) for classifying jamming signals based on their visual representation. Many GNSS jamming detectors generate spectral data in the form of power spectral density plots, spectrum snapshots, and waterfall diagrams, which capture distinctive characteristics of different jamming signals. By leveraging ML-based image recognition, it is possible to automatically categorize jamming signals and associate them with specific jammer models, thus enabling better tracking and prediction of interference sources. Machine learning-based image recognition has already been successfully applied in various fields, such as quality management and medical diagnostics, to automate complex analytical tasks [12, 13, 14].

This study represents the first step in applying similar techniques to GNSS interference detection. By determining whether ML can reliably classify commonly encountered jamming signals, this research lays the foundation for future developments in automated GNSS interference monitoring and prediction.

2. Data and Classification Methodology

A large dataset of jamming data was required to verify the feasibility of jamming signal classification via ML. This section first introduces how the jamming data were obtained. Then, it describes the ML network used for image recognition and justifies its selection. Finally, the evaluation methodology for determining the correct jamming signal category is given.

2.1 GNSS Jamming Data

GNSS jamming events data were captured during a long-term monitoring campaign lasting from May 2021 to March 2022. The campaign used the GSS100D GPS L1 jamming detector developed by Nottingham Scientific Limited (NSL), currently operating as GMV NSL. The detector was installed at the Czech busiest highway D1 which has an average traffic intensity of around 35 thousand vehicles per 24 hours at the location where the detector was installed [15].

Over the course of the monitoring campaign, 2069 jamming events were captured. The GSS100D detector has a monitoring bandwidth of 16 MHz and provides a spectrum snapshot and waterfall diagram for each of the jamming events. The spectrum snapshot shows the signal at its peak signal power level over the duration of the event whereas the waterfall diagram visualises a 100 μs long signal capture, 50 μs before and 50 μs after the peak power. Fig. 1 shows an example of the spectrum snapshot and the waterfall diagram for a randomly chosen jamming event.

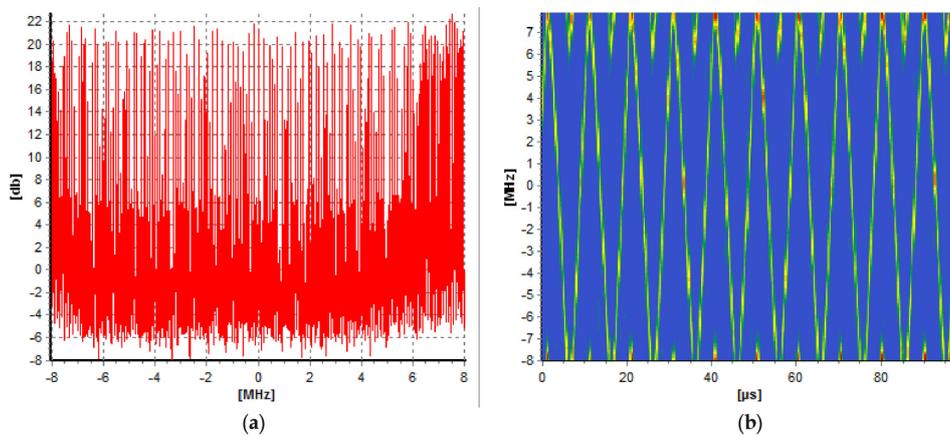


Fig. 1 A showcase of the GSS100D visual outputs, in (a) a spectrum snapshot and in (b) a waterfall diagram.

2.1.1 Jamming Signal Categories

As mentioned in Section 1, there is no official jamming signal categorization in place at the moment (year 2024). However, to validate the feasibility of an automatic

jamming signal categorization via an ML network a set of categories had to be adopted. The most commonly referred to set of jamming signal categories was defined within the STRIKE3 project [16]. Based on the distinctions laid out by the STRIKE3 project, seven of the following jamming signal categories were adopted for the analysis:

1. Single tone (ST): A spectrum snapshot shows a single dominant tone with high power. The waterfall diagram has a single dominant near vertical line in the region of affected frequency.
2. Multi tone (MT): A spectrum snapshot shows multiple distinct tones with high power at different frequencies. The waterfall diagram has multiple closely spaced near vertical lines in the region of affected frequency.
3. Wide sweep fast (WSF): A spectrum snapshot shows wide variation in power levels at all frequencies, often we see the shape of the reference spectrum defining the bottom edge of power levels. The waterfall diagram has clearly defined and separated linear or slightly curved diagonal lines across a wide frequency range, fast sweeps are characterised as having more than eight chirps per 100 μs .
4. Wide sweep slow (WSS): A spectrum snapshot shows wide variation in power levels at all frequencies, often we see the shape of the reference spectrum defining the bottom edge of power levels. The waterfall diagram has clearly defined and separated linear or slightly curved diagonal lines across a wide frequency range, slow sweeps are characterised as having two to seven chirps per 100 μs .
5. Narrow sweep (NS): A spectrum snapshot shows an increase in power levels across a narrow frequency range. The waterfall diagram has clearly defined and separated linear (or slightly curved) diagonal lines covering a small frequency range.
6. Sawtooth: A spectrum snapshot shows raised power over the affected frequency range. The waterfall diagram has linear sweeps in frequency across a wide range, decreases as well as increases in frequency with time, and the gradient of the downward slope is much sharper than the main upward slope.
7. Triangular: A spectrum snapshot shows raised power over an affected frequency range. The waterfall diagram has a clear decrease and increase in frequency with time, and the gradient and the power level of the downward and upward slopes are more equal than in the sawtooth case.

Fig. 2 displays seven spectrum snapshots and waterfall diagrams from the GSS100D detector, one for each of the seven jamming signal categories adopted. The images in Fig. 2 are laid out in the same order from the top left corner as in the previous bullet point list.

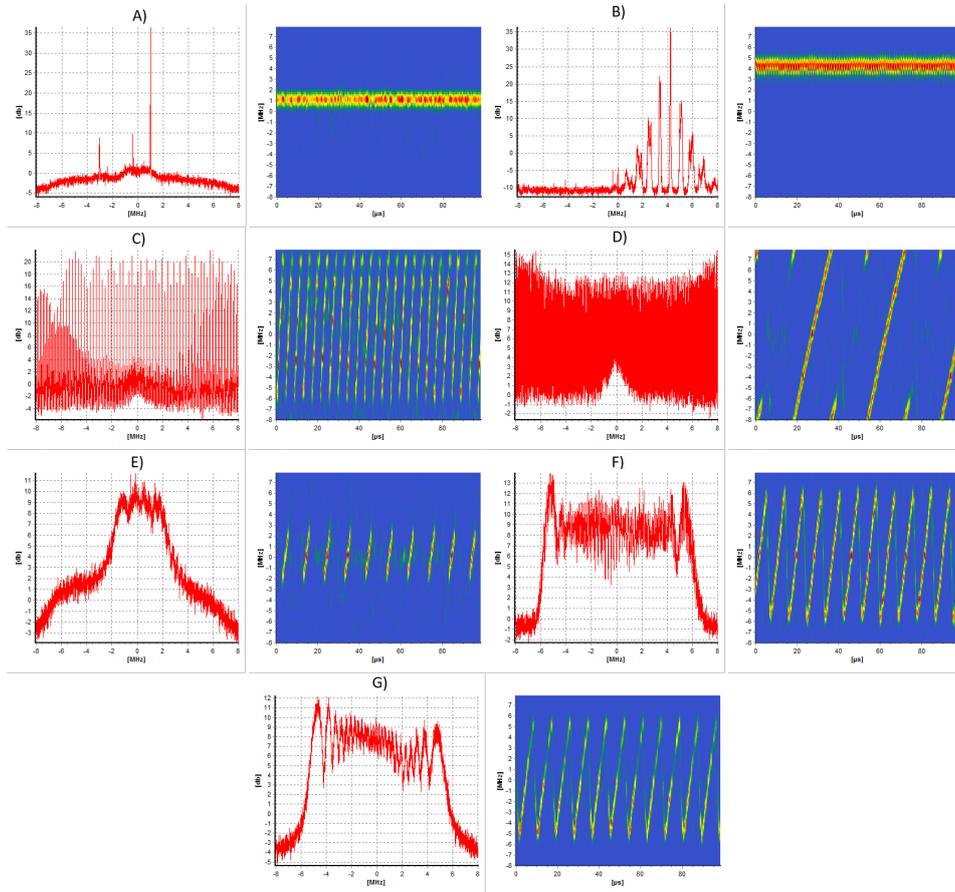


Fig. 2 Seven spectrum snapshots and waterfall diagrams from the GSS100D detector, each example representing one jamming signal category: 1. Single tone (ST), 2. Multi tone (MT), 3. Wide sweep fast (WSF), 4. Wide sweep slow (WSS), 5. Narrow sweep (NS), 6. Sawtooth, 7. Triangular.

2.1.2 Jamming Data Processing

Before being used for training and validation in the ML network the spectrum snapshots and waterfall diagrams from the GSS100D detector were processed as follows. First, the jamming signal category was manually determined by two human experts who labelled each image. Only images with identical labels from both experts were included in the dataset.

After labelling, the dataset was limited to 100 images in each of the aforementioned jamming signal categories, making a total of 700 data samples. The uniform limit of 100 images per category was chosen to ensure comparable success rate categorisation between individual categories.

Out of the 100 images in each category, 75 of them were used as training data for the model, and the remaining 25 images served as testing data to evaluate

correct category determination. The selection of training and testing data was randomized. Before being used in the analysis, images were cropped to remove their axes, event identification number, and the company logo to ensure only the visual representation of the jamming signal was taken into account by the ML model.

2.2 Residual Neural Network

The input data were images in PNG format from the professional GNSS detector, model GSS100D as showcased in Fig. 1. The text parts of the images were cropped and the L_p normalization was performed during the preprocessing. The L_p normalization is defined by Eq. 1, where \mathbf{v} is the normalized vector. The default values of $\epsilon = 10^{-12}$ and $p = 2$ were used in the model. Python library PyTorch was used for the training and evaluation of the neural network models [17].

$$\mathbf{v} = \frac{\mathbf{v}}{\max(\|\mathbf{v}\|_p, \epsilon)} \quad (1)$$

Various machine learning and neural network approaches in the signal spectrogram or scalogram classifications can be found in the literature. One of them is AlexNet. AlexNet is a convolutional neural network comprising eight layers—five convolutional layers followed by three fully connected layers—using Rectified Linear Unit (relu) activations and overlapping max pooling to capture complex features. It employs data augmentation, dropout, and local response normalization (LRN) to improve generalization and mitigate overfitting [18, 19]. Another heavily used approach is VGG. VGG is a deep convolutional network characterized by its use of a uniform architecture with small 3×3 convolutional filters throughout the network. It consists of sequential convolutional layers, followed by max pooling and fully connected layers, maintaining a simple and consistent design that promotes depth over complexity. The network relies on relu activations and employs a large number of parameters, which necessitates careful regularization and substantial computational resources for training [20]. EfficientNet is another widely used approach. EfficientNet is a family of convolutional neural networks that employ a compound scaling method to uniformly scale depth, width, and resolution, achieving improved accuracy and efficiency. It leverages mobile inverted bottleneck convolution layers, swish activations, and squeeze-and-excitation blocks to enhance feature representation [21]. Another machine learning approach which was selected by the authors of this paper is ResNet and it is described in greater detail later in this section.

Numerous studies employ one or more of these approaches for classification tasks when processing spectrograms in signal processing. Li and others use the methods to classify spectrograms of jamming signals of orthogonal frequency division multiplexing receivers [22]. Merh and Dervis use AlexNet and ResNet to classify GNSS jamming signals. However, they do not compare the performance of ResNet-50 with other versions [23]. A similar task is solved by AlexNet, VGG, and ResNet by Elango and others [24]. This paper also lacks comparison of ResNet variants and uses only ResNet-18. The GNSS jamming signal classification can be also solved by VGG as has been proved by Swinney and Woods [25]. Zhang and Krunch [26] use AlexNet, VGG, ResNet, and EfficientNet to classify the jamming

of Wi-Fi signal. Ujan and others [27] use AlexNet, VGG, and ResNet to classify radio frequency interference. Xu and others [28] use AlexNet to tackle the interference classification problem for the frequency hopping communication system. Bhatti and others [29] use AlexNet, VGG, and ResNet to identify various radio technologies and their associated interferences.

ResNet is an architecture of convolution neural networks that aims to address the degradation problem encountered when training very deep neural networks. As network depth increases, traditional architectures often suffer from vanishing or exploding gradients, leading to higher training and test errors [30]. The key innovation of ResNet was the use of residual learning through skip connections, which allow layers to learn residual functions with respect to the inputs of preceding layers, as can be seen in Fig. 3. Specifically, identity mappings bypass one or more layers and directly feed the input to subsequent layers, effectively creating shortcut paths for gradient flow. This approach alleviates the vanishing gradient problem [30].

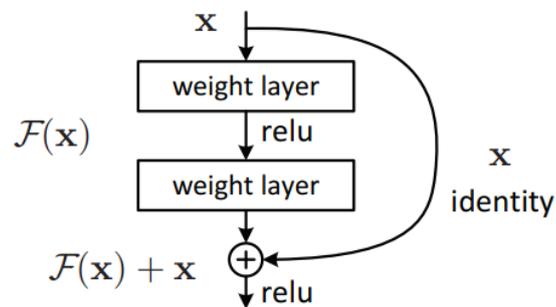


Fig. 3 Block diagram of the skip connections technique implemented within the ResNet architecture. See Eq. 2 for relu definition.

The ResNet network generally provides strong performance, scalability, ability to solve general problems and efficient training. There are multiple variants of the ResNet network available differing in the number of network layers. The version with the least layers is ResNet-18 [30].

Examples of successful use cases applying the ResNet network on datasets similar to the GSS100D outputs, such as audio spectrogram [31, 32], radar spectrogram [33], or accelerometer signals [34], can be found in the literature. However, ResNet also performs strongly in tasks focused on line or curve shape recognition, such as handwritten text [35], and scene or document text [36]. As the curve shape is the key to recognition of the jamming category, ResNet is expected to perform well on the spectrum snapshots and waterfall diagrams.

To give an example of the ResNet network architecture, the ResNet-18 architecture is displayed in Fig. 4. The ResNet-18 model consists of 18 layers, organized into four main stages of residual blocks. Each residual block is composed of two 3×3 convolution layers with batch normalization and relu activation. Relu activation is defined by the Eq. 2.

$$\text{relu}(x) = (x)^+ = \max(0, x) \quad (2)$$

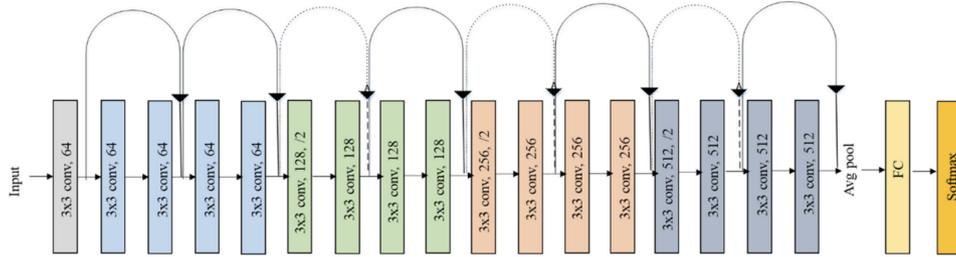


Fig. 4 ResNet-18 architecture layers with highlighted skip connections and residual blocks differentiated by colour [37].

Notably, downsampling occurs in the second block of each stage via a stride of two, facilitating the reduction of spatial dimensions while increasing the number of filters. The final layers include an average pooling layer, a fully connected (FC) layer, and a softmax output for classification [37].

In the case presented in this study, the model outputs seven classification labels, as there are seven different groups of jamming signals in the dataset. An overview of all ResNet architectures across all the models with different number of layers tested within the study can be seen in Fig. 5.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10 ⁹	3.6×10 ⁹	3.8×10 ⁹	7.6×10 ⁹	11.3×10 ⁹

Fig. 5 Architecture and computational complexity of ResNet variants (18, 34, 50, 101, and 152 layers) [30].

The hardware used for all experiments was a MacBook Pro with an Apple M3 Max, CPU with 14 cores, 30-core GPU, and 16-core Neural Engine with 36 GB shared memory. As this study aims to verify the feasibility of deep-learning-based GNSS jamming signals categorizations, it does not have the ambition to fine tune each model separately or to provide a fully hyperparameter-optimized model. Rather than that, the study strives to compare the performance of various ResNet and evaluates whether the image recognition approach is beneficial for the GNSS jamming incident categorization.

Additionally, it should adjudicate whether a shallower network would provide an acceptable prediction model or whether deeper networks should be used. A fixed number of 50 epochs was set for all models to achieve the goal of the study. The hyperparameters of the model were set based on the documentation of PyTorch and based on generally recommended hyperparameters ranges and values. The batch size was set to five because of the size of the dataset. The number of subprocesses was set to four, and the learning rate was set to $1 \cdot 10^{-3}$. StepLR was used for learning rate scheduling with the learning rate decay parameter set to 0.1 and Stochastic Gradient Descent (SDG) was used as the optimizer with momentum set to 0.9.

2.3 Evaluation

The success rate of the ResNet models for classification was evaluated via a confusion matrix [38] showing both the number of true and false classifications. Additionally, the loss curves and accuracy curves are showcased to visualize how the network is learning and performing over time [39].

The epoch loss can be calculated using Eq. 3, where N is the total number of samples in the current phase (training or validation) and L_i is the loss in the i -th sample, calculated by the cross-entropy loss [40].

$$\mathcal{L}_{\text{epoch}} = \frac{1}{N} \sum_{i=1}^N L_i \quad (3)$$

The accuracy is calculated using Eq. 4, where y_i is the true label for the i -th sample, \hat{y}_i is the predicted label for the i -th sample, and \mathbb{I} is the indicator function which is equal to one if the comparison is true or equal to zero if the comparison is false.

$$\text{Acc}_{\text{epoch}} = \frac{1}{N} \sum_{i=1}^N \mathbb{I}(y_i = \hat{y}_i) \quad (4)$$

3. Results

First, the ResNet classification performance in the form of a confusion matrix is given for each ResNet model variant which differs in the number of layers. The confusion matrixes show cumulative results of all 4 cross validation runs, meaning the total number of classifications in a single row is 4×25 since there were 25 testing images in each jamming signal category. Second, loss curves and accuracy curves are given for each ResNet variant for further performance description.

3.1 ResNet-18

For the ResNet-18, Tab. I showcases the confusion matrix classification success rate and Fig. 6 displays the loss and accuracy curves. Overall, ResNet-18 showed reasonably strong performance, particularly in validation, where accuracy reached over 81% by later epochs (and almost 90% for three of the four subsets). The validation loss decreased consistently from 1.47 to around 0.3. This suggests that despite being a shallower network, ResNet-18 efficiently captured patterns in the data, making it a strong candidate for fast training or resource-limited scenarios.

	ST	MT	WSF	WSS	NS	Sawtooth	Triangular
ST	96	4	0	0	0	0	0
MT	11	85	0	0	4	0	0
WSF	0	0	96	0	1	2	1
WSS	0	0	1	96	0	0	3
NS	0	7	2	2	86	2	1
Sawtooth	0	0	1	0	2	91	6
Triangular	0	0	0	1	0	5	94

Tab. I Confusion matrix showing both the correctly classified (highlighted in green) and misclassified jamming events for ResNet-18.

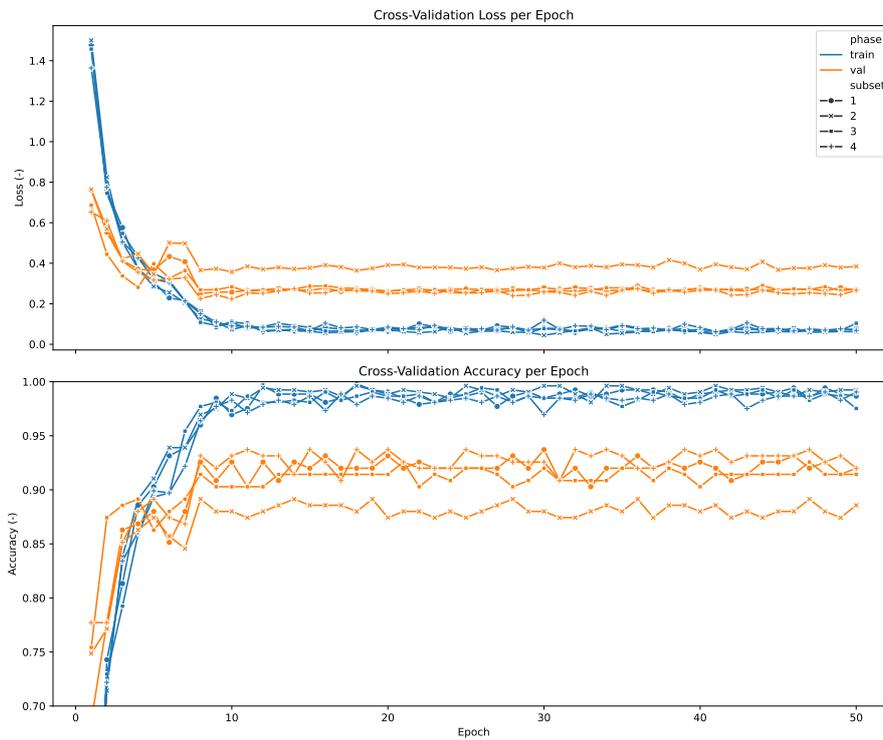


Fig. 6 Loss and accuracy curves showcasing the performance of the ResNet-18.

3.2 ResNet-34

For the ResNet-34, Tab. II showcases the confusion matrix classification success rate and Fig. 7 displays the loss and accuracy curves. ResNet-34 showed a relatively higher loss during the initial epochs, with training loss starting at 1.85 and dropping to 0.46 after 50 epochs. Validation accuracy lagged behind the deeper networks, initially showing high variability between epochs. The training accuracy stabilized around 80%, slightly underperforming compared to the shallower

ResNet-18. A potential update of network hyperparameters that may increase the accuracy of ResNet-34 are summarized in the Section 4. The hyperparameters optimization for various ResNet-34 variants is beyond scope of this paper and may be a goal of additional research.

	ST	MT	WSF	WSS	NS	Sawtooth	Triangular
ST	90	9	0	1	0	0	0
MT	39	56	0	0	5	0	0
WSF	0	0	89	1	0	1	9
WSS	0	0	1	93	2	0	4
NS	2	3	2	11	79	3	0
Sawtooth	0	0	4	1	1	87	7
Triangular	0	0	0	3	5	8	84

Tab. II Confusion matrix showing both the correctly classified (highlighted in green) and misclassified jamming events for ResNet-34.

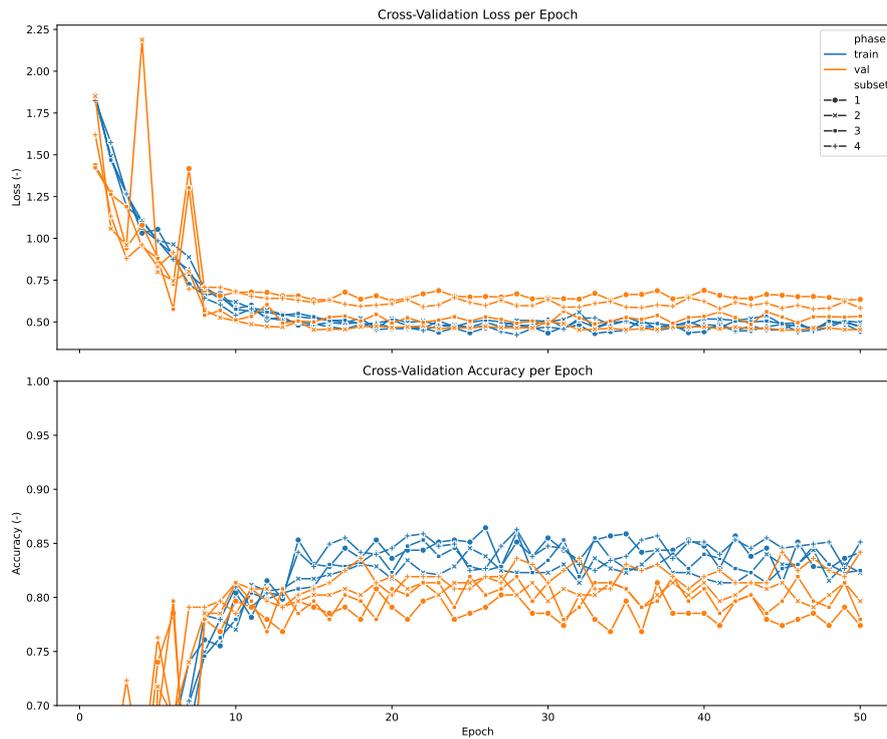


Fig. 7 Loss and accuracy curves showcasing the performance of the ResNet-34.

3.3 ResNet-50

For the ResNet-50, Tab. III showcases the confusion matrix classification success rate and Fig. 8 displays the loss and accuracy curves. In the case of ResNet-50, training loss exhibited a steady decrease from 1.55 to values around 0.6 for the first two subsets and around 0.2 for the remaining two subsets. Validation accuracy was slightly better in comparison to ResNet-18 and stable across all four cross validations subsets.

	ST	MT	WSF	WSS	NS	Sawtooth	Triangular
ST	96	4	0	0	0	0	0
MT	9	86	0	0	5	0	0
WSF	0	0	94	1	0	3	2
WSS	0	0	1	95	2	0	2
NS	0	4	2	0	91	3	0
Sawtooth	0	0	1	0	2	91	6
Triangular	0	0	2	2	1	7	88

Tab. III Confusion matrix showing both the correctly classified (highlighted in green) and misclassified jamming events for ResNet-50.

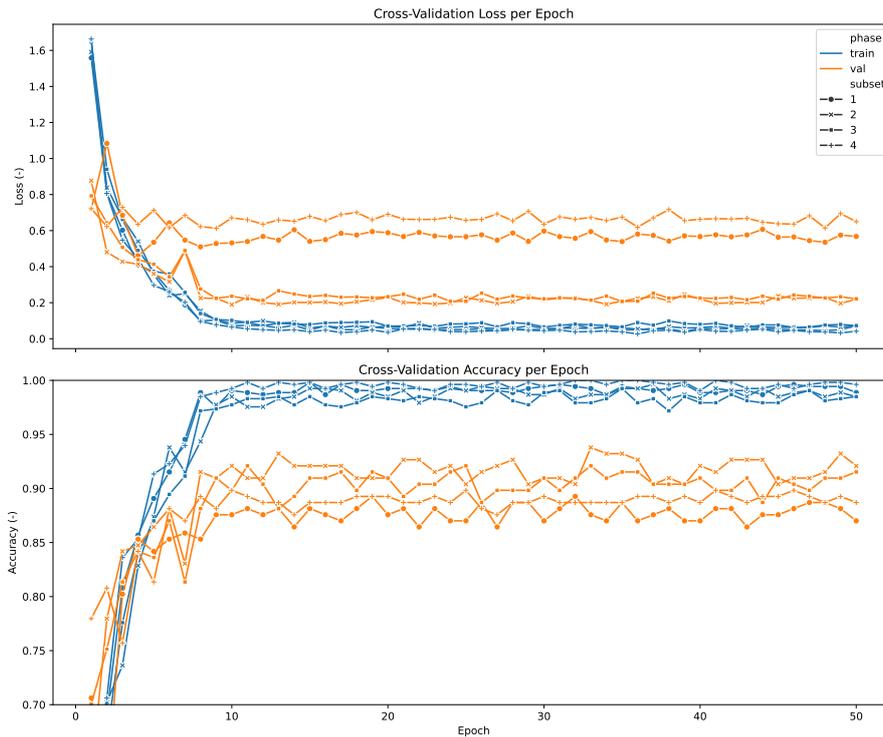


Fig. 8 Loss and accuracy curves showcasing the performance of the ResNet-50.

3.4 ResNet-101

For the ResNet-101, Tab. IV showcases the confusion matrix classification success rate and Fig. 9 displays the loss and accuracy curves. ResNet-101 demonstrated reliable training, with initial training loss starting at 1.55 and dropping to values near 0.5 for the first two subsets and 0.2 for the remaining two subsets. Validation accuracy followed an upward trend, achieving over 85% during the second half of the training.

	ST	MT	WSF	WSS	NS	Sawtooth	Triangular
ST	97	3	0	0	0	0	0
MT	11	84	0	0	5	0	0
WSF	0	0	95	1	0	3	1
WSS	0	0	2	97	0	0	1
NS	0	6	1	2	88	2	1
Sawtooth	0	0	3	0	1	89	7
Triangular	0	0	1	1	1	5	92

Tab. IV Confusion matrix showing both the correctly classified (highlighted in green) and misclassified jamming events for ResNet-101.

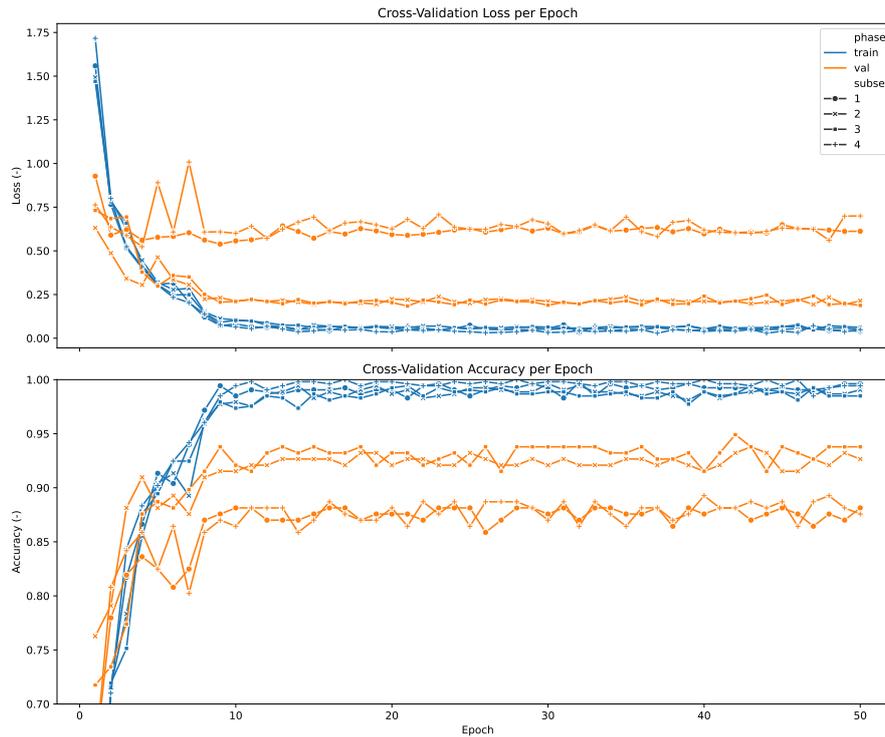


Fig. 9 Loss and accuracy curves showcasing the performance of the ResNet-101.

3.5 ResNet-152

For the ResNet-152, Tab. V showcases the confusion matrix classification success rate and Fig. 10 displays the loss and accuracy curves. ResNet-152 exhibited strong performance throughout training, with a steady decline in loss and increasing accuracy across epochs. Notably, validation accuracy consistently remained above 75% after the initial epochs, reaching peaks of over 90% outperforming the remaining networks. This indicates that the network effectively learned from the data while maintaining a good balance between accuracy and loss, particularly in the validation phases.

	ST	MT	WSF	WSS	NS	Sawtooth	Triangular
ST	98	8	0	0	0	0	0
MT	14	83	0	0	3	0	0
WSF	0	0	98	1	0	1	
WSS	0	0	0	96	1	1	2
NS	0	5	1	2	90	2	0
Sawtooth	0	0	1	0	1	94	4
Triangular	0	0	1	1	0	10	88

Tab. V Confusion matrix showing both the correctly classified (highlighted in green) and misclassified jamming events for ResNet-152.

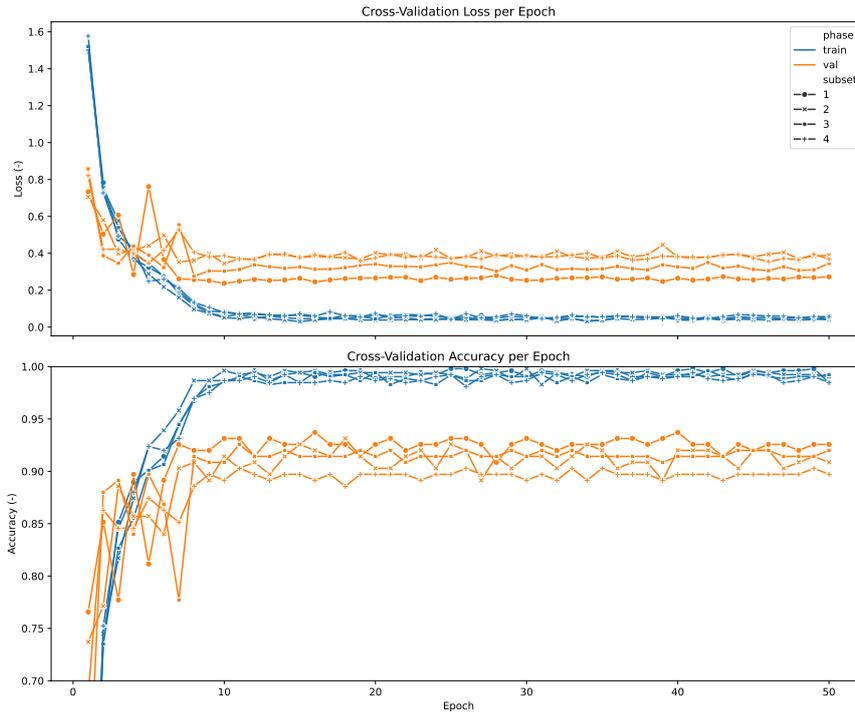


Fig. 10 Loss and accuracy curves showcasing the performance of the ResNet-152.

3.6 Statistical Comparison of ResNet Variants

The Friedman test was used to compare the accuracy metric values provided by various ResNet variants. The Friedman test is a non-parametric version of ANOVA and it can be used to compare average values of more than 2 data samples and it can be used in repeated measurement designs [41]. The Friedman test because it cannot be assumed the accuracy metric values meet the assumption of the ANOVA test. There are 4 accuracy values for each ResNet variant as the data were split into 4 subsets. The hypotheses of the test are as follows:

- The null hypothesis: The average accuracy of all ResNet variants are equal.
- There is at least one difference between the average accuracy of ResNet variants.

Accuracy metric values for each ResNet variant are depicted in Tab. VI. The test is performed with $\alpha = 5$ level of significance. The statistics of the test is 8.468 and the p-value is 0.076. It means that we cannot reject the null hypothesis and we did not prove the difference in average values of the ResNet variants.

	Subset 1	Subset 2	Subset 3	Subset 4
ResNet-18	0.9371	0.9371	0.8914	0.9257
ResNet-34	0.8136	0.8192	0.8192	0.8418
ResNet-50	0.8983	0.9209	0.9379	0.8927
ResNet-101	0.8814	0.9492	0.9379	0.8927
ResNet-152	0.9371	0.9029	0.9257	0.9314

Tab. VI Accuracy metric values for the ResNet variants.

4. Discussion

When comparing the determination success rates of individual jamming signal categories, the most commonly interchanged categories were Single tone and Multi tone. Although ST and MT differ in the spectrum snapshot, they might be very similar in the waterfall diagram. This similarity probably leads to the significant misclassification being noticeable across all ResNet model variants. Less pronounced but also constant across ResNet model variants case of interchanged category classification can be seen between Sawtooth and Triangular, both of which are commonly observed in commercial jammers. Finally, the Narrow sweep category was in about 5% of the cases misclassified as Multi tone, except in ResNet-34 where it was misclassified as the Wide sweep slow category in 11% of the cases.

The results in Section 3 highlight that the deeper networks ResNet-152 and ResNet-101 generally perform better in terms of both validation accuracy and loss, reflecting their ability to capture more detailed features of the data. However, ResNet-18, despite being much shallower, offers competitive accuracy, particularly when considering the trade-off between model complexity and performance. Specifically, the ResNet-152 required approximately quadruple the time to advance to

another epoch compared to ResNet-18. Moreover, ResNet-18 outperforms ResNet-34, which struggled to maintain competitive performance, showing higher loss values and lower validation accuracy, potentially due to underfitting or limited model capacity. These promising results provide the necessary feasibility for better jammer classification which is a key element of individual jammed recognition and proactively tool development. Additionally, the performance differences between ResNet-152 and ResNet-18 can be measured in units of percentage points. It is reasonable to assume that with hyperparameter tuning, a model based on ResNet-18 would achieve accuracy slightly above 90%.

Regarding the hyperparameters tuning, several hyperparameters could be tuned. A change in the learning rate would change the size of updates of the models' weight. Another optimizer, e.g., Adam, could be used instead of SGD or the momentum of SGD could be changed. Batch size would be limited by the limited size of the dataset, however, this hyperparameter could be also optimized, keeping in mind the dataset size validation. Additionally, optimization of weight decay would affect the speed of convergence and could be used to prevent overfitting. The training time of the models may be reduced by freezing the initial layers, which may not cause a loss of accuracy and could reduce the training time up to 20% [42]. Dropout could be used to prevent overfitting [43]. The number of epochs could also be optimized, as both epoch loss and accuracy seem to require fewer than 50 epochs to converge to stable values [39, 44, 43].

Additional improvements may be achieved through further data transformation. Different neural network models may be useful for spectral snapshot diagrams and waterfall diagrams. The spectral snapshot could be converted to numerical values. Processing of numerical values with a different type of neural network might achieve better performance, even though ResNet, in general, performs well in the curve-shape recognition tasks [35, 36]. Several online services and software packages can generate numerical values from a provided plot, for example, WebPlotDigitizer, Engauge Digitizer, or Python library plotdigitizer. As using these services and libraries would require fine-tuning and validation of the results, it was not included in the current paper and can be a subject of additional research.

On the other hand, the image recognition convolution neural network seems to work well with the waterfall plot. The performance of a model based solely on the waterfall could be also tested, as Fig. 2 demonstrates the significant differences among jamming categories (Sawtooth and Triangular categories can be considered as the only exceptions). So using two deep learning models and merging their results may increase the accuracy of the model. The output of both models can be concatenated and sent to a final fully connected classification layer to gain the final classification result.

5. Conclusion

This study demonstrated the feasibility of using deep learning, specifically ResNet architectures, to categorize GNSS jamming signals with high accuracy. The findings highlight the potential for machine learning-based image recognition to enhance GNSS interference detection and classification, offering a scalable and automated alternative to manual analysis. By achieving over 90% classification accuracy in

most tested models, this approach lays the groundwork for more advanced, proactive interference mitigation strategies.

The significance of this work extends beyond classification—it provides a foundation for developing predictive models that could anticipate jamming events based on historical patterns. Such advancements could improve GNSS resilience in critical infrastructure, transportation, and security applications, helping authorities and service providers mitigate the impact of intentional jamming more effectively. Although the classification of the Sawtooth and Triangula categories which are commonly seen in commercial jammers might pose a challenge, there are many ways of further optimising the model as described in Section 4. The follow-up development would include a feasibility verification of individual jammer model recognition. A more comprehensive dataset with multiple jammer models included via deliberate experiments will be required for the next development step.

The study was limited to a single type of GNSS jamming detector. However, other professional GNSS jamming detectors also provide visual outputs or diagrams representing the jamming signal meaning the usage of the presented ResNet model might be universal across multiple detector models. However, the use of the network requires a large training dataset which might require months to gather.

Another use case for the ResNet network utilized in this study might include a design of new jamming signal categories via unsupervised learning. Since there are no set standards for this categorisation, there is room for new proposals.

Acknowledgement

The work presented was supported by the Czech Technical University school grant ref. SGS23/138/OHK2/2T/16. The GPS interference data were obtained within a NAVISP element 3 project “GNSS Vulnerability and Mitigation in Czech Republic”.

References

- [1] EUSPA. *EUSPA EO and GNSS Market Report.2024 / Issue 2* . 2024. Available also from: <https://data.europa.eu/doi/10.2878/73092>. ISBN 978-92-9206-079-4.
- [2] RTCA SC-159. *Assessment of Radio Frequency Interference Relevant to the GNSS*. 2008.
- [3] U.S. SPACE FORCE. *Space and Missile Systems Center: NAVSTAR GPS Space Segment/Navigation User Interfaces, GPS Interface Specification IS-GPS-200, Revision M*, U.S. Space Force, 2021. Technical Report.
- [4] VOLPE J.A. *Vulnerability assessment of the transportation infrastructure relying on global positioning system*. 2001. Available also from: <https://rosap.ntl.bts.gov/view/dot/8435>.
- [5] WHITTY C., WALPORT M. *Satellite-Derived Time and Position: A Review*. UK Government, 2018. Technical Report.

- [6] BARTL S., KADLETZ M., BERGLEZ P., DUŠA T. Findings from Interference Monitoring at a European Airport. In: *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2021)*, 2021. Available also from: <https://doi.org/10.33012/2021.17975> doi: 10.33012/2021.17975.
- [7] STANISAK M., von HÜNERBEIN K., BESTMANN U., LANGE W. Measured GNSS Jamming Events at German Motorways. In: *Measured GNSS Jamming Events at German Motorways*, 2016.
- [8] RØDNINGSBY A., MORRISON A., SOKOLOVA N., GERRARD N., ROST C. RFI Monitoring of GNSS Signals on Norwegian Highways. In: *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, 2020, pp. 3536–3549. doi: 10.33012/2020.17671.
- [9] ELIARDSSON P., ALEXANDERSSON M., PATTINSON M., HILL S., WAERN A., YING Y., FRYGANIOTIS D. Results from measuring campaign of electromagnetic interference in GPS L1-band. In: *2017 International Symposium on Electromagnetic Compatibility - EMC EUROPE*, 2017. Available also from: <http://dx.doi.org/10.1109/EMCEurope.2017.8094660> doi: 10.1109/emceurope.2017.8094660.
- [10] THOMBRE S., BHUIYAN M.Z.H., ELIARDSSON P., GABRIELSSON B., PATTINSON M., DUMVILLE M., FRYGANIOTIS D., HILL S., MANIKUNDALAM V., PÖLÖSKEY M., LEE S., RUOTSALAINEN L., SÖDERHOLM S., KUUSNIEMI H. GNSS Threat Monitoring and Reporting: Past, Present, and a Proposed Future. *Journal of Navigation*. 2017, 71(3), pp. 513–529, doi: 10.1017/s0373463317000911. ISSN 1469-7785.
- [11] EU-U.S. COOPERATION ON SATELLITE NAVIGATION, WG-C SERVICE RESILIENCE SUB-GROUP. *Radio Frequency Interference Monitoring Measures and Systems*. 2020. Technical Report. Available also from: <https://www.gps.gov/policy/cooperation/europe/2016/working-group-c/ARAIM-milestone-3-report.pdf>.
- [12] LI Z., ZHOU D., WAN L., LI J., MOU W. Heartbeat classification using deep residual convolutional neural network from 2-lead electrocardiogram. *Journal of Electrocardiology*. 2020, 58, pp. 105–112, doi: 10.1016/j.jelectrocard.2019.11.046. ISSN 0022-0736.
- [13] GAO M., QI D., MU H., CHEN J. A Transfer Residual Neural Network Based on ResNet-34 for Detection of Wood Knot Defects. *Forests*. 2021, 12(2), pp. 212, doi: 10.3390/f12020212. ISSN 1999-4907.
- [14] ZHANG H., YANG G., HUANG Y., YUAN F., YIN Y. Multi-Scale and Attention based ResNet for Heartbeat Classification. In: *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021. Available also from: <http://dx.doi.org/10.1109/ICPR48806.2021.9413052> doi: 10.1109/icpr48806.2021.9413052.
- [15] ŘEDITELSTVÍ SILNIC A DÁLNIC ČR. *Sčítání dopravy*. 2016. [Accessed 2021-6-24]. Available also from: <https://www.rsd.cz/wps/portal/web/Silnice-a-dalnice/Scitani-dopravy>.
- [16] TOWLSON O., PAYNE D., ELIARDSSON P., VENKATESH M., PATTINSON M., DUMVILLE M. *STRIKE3 D6.2: THREAT DATABASE ANALYSIS REPORT*. Nottingham Scientific Limited (NSL), 2019. Project Deliverable.

- [17] PASZKE A., GROSS S., MASSA F., LERER A., BRADBURY J., CHANAN G., KILLEEN T., LIN Z., GIMELSHEIN N., ANTIGA L., DESMAISON A., KÖPF A., YANG E., DEVITO Z., RAISON M., TEJANI A., CHILAMKURTHY S., STEINER B., FANG L., BAI J., CHINTALA S. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 2019. Available also from: <https://arxiv.org/abs/1912.01703>.
- [18] KRIZHEVSKY A., SUTSKEVER I., HINTON G.E. ImageNet Classification with Deep Convolutional Neural Networks. In: F. PEREIRA, C. BURGESS, L. BOTTOU, K. WEINBERGER, eds. *Advances in Neural Information Processing Systems*, 2012. Available also from: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.
- [19] ALOM M.Z., TAHA T.M., YAKOPCIC C., WESTBERG S., SIDDIKE P., NASRIN M.S., VAN ESESN B.C., AWWAL A.A.S., ASARI V.K. The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*. 2018.
- [20] SIMONYAN K., ZISSERMAN A. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. Available also from: <https://arxiv.org/abs/1409.1556>.
- [21] TAN M., LE Q.V. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. 2020. Available also from: <https://arxiv.org/abs/1905.11946>.
- [22] LI Y., PAWLAK J., PRICE J., AL SHAMAILEH K., NIYAZ Q., PAHEDING S., DEVABHAKTUNI V. Jamming detection and classification in OFDM-based UAVs via feature-and spectrogram-tailored machine learning. *Ieee Access*. 2022, 10, pp. 16859–16870.
- [23] MEHR I.E., DOVIS F. A Deep Neural Network Approach for Classification of GNSS Interference and Jammer. *IEEE Transactions on Aerospace and Electronic Systems*. 2024, pp. 1–18, doi: [10.1109/TAES.2024.3462662](https://doi.org/10.1109/TAES.2024.3462662).
- [24] ELANGO A., UJAN S., RUOTSALAINEN L. Disruptive GNSS signal detection and classification at different power levels using advanced deep-learning approach. In: *2022 International Conference on Localization and GNSS (ICL-GNSS)*, 2022, pp. 1–7.
- [25] SWINNEY C.J., WOODS J.C. GNSS Jamming Classification via CNN, Transfer Learning & the Novel Concatenation of Signal Representations. In: *2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*, 2021, pp. 1–9. doi: [10.1109/CyberSA52016.2021.9478250](https://doi.org/10.1109/CyberSA52016.2021.9478250).
- [26] ZHANG Z., KRUNZ M. Detection and Classification of Smart Jamming in Wi-Fi Networks Using Machine Learning. In: *MILCOM 2023-2023 IEEE Military Communications Conference (MILCOM)*, 2023, pp. 919–924.
- [27] UJAN S., NAVIDI N., JR LANDRY R. An Efficient Radio Frequency Interference (RFI) Recognition and Characterization Using End-to-End Transfer Learning. *Applied Sciences*. 2020, 10(19), doi: [10.3390/app10196885](https://doi.org/10.3390/app10196885). ISSN 2076-3417.
- [28] XU C., REN J., YU W., JIN Y., CAO Z., WU X., JIANG W. Time-frequency analysis-based deep interference classification for frequency hopping system. *EURASIP Journal on Advances in Signal Processing*. 2022, 2022(1), pp. 90.
- [29] BHATTI F.A., KHAN M.J., SELIM A., PAISANA F. Shared Spectrum Monitoring Using Deep Learning. *IEEE Transactions on Cognitive Communications and Networking*. 2021, 7(4), pp. 1171–1185, doi: [10.1109/TCCN.2021.3071149](https://doi.org/10.1109/TCCN.2021.3071149).

- [30] HE K., ZHANG X., REN S., SUN J. Deep Residual Learning for Image Recognition. *CoRR*. 2015, abs/1512.03385.
- [31] SANKUPELLAY M., KONOVALOV D. Bird call recognition using deep convolutional neural network, ResNet-50. In: *Proc. Acoustics*, 2018, 2018, pp. 1–8.
- [32] REN Y., LIU D., XIONG Q., FU J., WANG L. Spec-resnet: a general audio steganalysis scheme based on deep residual network of spectrogram. *arXiv preprint arXiv:1901.06838*. 2019.
- [33] PARK D., LEE S., PARK S., KWAK N. Radar-spectrogram-based UAV classification using convolutional neural networks. *Sensors*. 2020, 21(1), pp. 210.
- [34] SÁNCHEZ S.H., POZO R.F., GÓMEZ L.A.H. Driver identification and verification from smartphone accelerometers using deep neural networks. *IEEE Transactions on Intelligent Transportation Systems*. 2020, 23(1), pp. 97–109.
- [35] JAYAKANTHAN R., KUMAR A.H., SANKARRAM N., CHARULATHA B., RAMESH A. Handwritten tamil character recognition using ResNet. *International Journal of Research in Engineering, Science and Management*. 2020, 3(3), pp. 133–137.
- [36] DIAZ D.H., QIN S., INGLE R., FUJII Y., BISSACCO A. Rethinking text line recognition models. *arXiv preprint arXiv:2104.07787*. 2021.
- [37] RAMZAN F., KHAN M.U.G., REHMAT A., IQBAL S., SABA T., REHMAN A., MEHMOOD Z. A deep learning approach for automated diagnosis and multi-class classification of Alzheimer's disease stages using resting-state fMRI and residual neural networks. *Journal of medical systems*. 2020, 44, pp. 1–16.
- [38] DÜNTSCH I., GEDIGA G. Confusion Matrices and Rough Set Data Analysis. *Journal of Physics: Conference Series*. 2019, 1229(1), pp. 012055, doi: [10.1088/1742-6596/1229/1/012055](https://doi.org/10.1088/1742-6596/1229/1/012055). ISSN 1742-6596.
- [39] GOODFELLOW I., BENGIO Y., COURVILLE A. *Deep Learning*. 2016. <http://www.deeplearningbook.org>.
- [40] MAO A., MOHRI M., ZHONG Y. *Cross-Entropy Loss Functions: Theoretical Analysis and Applications*. 2023. Available also from: <https://arxiv.org/abs/2304.07288>.
- [41] SHELDON M.R., FILLIYAW M.J., THOMPSON W.D. The use and interpretation of the Friedman test in the analysis of ordinal-scale data in repeated measures designs. *Physiotherapy Research International*. 1996, 1(4), pp. 221–228.
- [42] BROCK A., LIM T., RITCHIE J.M., WESTON N. FreezeOut: Accelerate Training by Progressively Freezing Layers. *ArXiv*. 2017, abs/1706.04983.
- [43] SRIVASTAVA N., HINTON G.E., KRIZHEVSKY A., SUTSKEVER I., SALAKHUT-DINOV R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 2014, 15, pp. 1929–1958.
- [44] AGGARWAL C.C. *Neural Networks and Deep Learning: A Textbook*. 1st ed. Cham Springer Nature, 2018. ISBN 9783319944630; 3319944630; 3319944622; 9783319944623;