Noise Power Density Estimation Based on Deep Learning Using Spectrograms Extracted from Wireless Signals

Myke D.M. Valadão[1,3], André L.A. da Costa[2], Éderson R. da Silva[2], Alexandre C. Mateus[2], Waldir S.S. Júnior[1]
[1]Federal University of Amazonas (UFAM), AM-Brazil
[2]Federal University of Uberlândia (UFU), MG-Brazil
[3]SiDi, AM-Brazil
Emails: mykedouglas@ufam.edu.br, alacosta@ufu.br, ersilva@ufu.br, acmateus@ufu.br, waldirjr@ufam.edu.br

***Abstract*— In communication systems, noise is almost invariably present, originating from a multitude of sources and variables. These sources include thermal effects, interference, quantization, and channel imperfections, contributing to the random nature of noise. Determining noise levels is crucial and remains a pervasive challenge in communication systems, especially in recent times when better utilization of spectrum sensing is required. In this paper, we propose a noise estimation method based on deep learning using spectrograms extracted from wireless signals. The proposed method achieved promising results using several state-of-art computer vision architectures.**

***Keywords*— Noise Power Density, Spectrogram, Deep Learning.**

I. Introduction

In communication systems, noise is almost invariably present, originating from a multitude of sources and variables. These sources include thermal effects, interference, quantization, and channel imperfections, contributing to the random nature of noise. Given the uncertainty surrounding its characteristics, noise poses a pervasive challenge in communication systems. Consequently, various types of noise can detrimentally affect signal quality, leading to a reduction in the signal-to-noise ratio (SNR). As the SNR decreases, both communication range and bandwidth become limited, compromising the dynamic efficiency and scalability of communication systems [1], [2].

Several variables can influence the increase of noise in communication systems. Factors such as temperature fluctuations and electromagnetic interference can heighten signal corruption by noise. Additionally, channel characteristics, including attenuation, dispersion, and multipath effects, distance between users, can exacerbate noise levels during signal transmission [1]. The combination of these variables underscores the complexity of managing noise in communication systems, ultimately leading to a drastic reduction in the quality of signal transmission and reception, thereby compromising the integrity and efficiency of the communication systems.

Determining noise levels is crucial and remains a pervasive challenge in communication systems, especially in recent times when better utilization of spectrum sensing is required [3]. Therefore, methods that estimate noise levels can aid in designing better communication systems. By adjusting transmitted power, allocating bandwidth, and optimizing frequency channels more efficiently, overall network performance and user experience can be significantly improved [4].

In this paper, we propose a noise estimation method based on deep learning using spectrograms extracted from wireless signals. Different levels of additive white Gaussian noise (AWGN) are introduced to the signals. Spectrograms are extracted from these signals and used as input for state-of-the-art computer vision models employed for classification and regression purposes. The models estimate the noise power density, specifically AWGN levels. The proposed method achieved excellent results, with an accuracy superior to 97% on the test dataset.

A. Contributions

The contributions of this paper can be summarized as follows: (1) In signal generation, higher levels of AWGN noise were inserted into the signal. Additionally, the user's location changes over time, and several variables influencing the signal are introduced; (2) The Hilbert transform is employed to highlight singular features in the signals; (3) Several state-of-the-art computer vision architectures were utilized to estimate the noise; and (4) The proposed method achieved high performance compared to similar proposals.

II. Related works

The authors in [5] introduced a novel method for estimating the signal-to-noise ratio (SNR). This approach leverages a deep learning network called DINet, which integrates a denoising convolutional neural network (DnCNN) with an image restoration convolutional neural network (IRCNN) operating in parallel. By utilizing the sounding reference signal, DINet achieves improved SNR estimation performance compared to existing algorithms. Evaluation of the method involved comparing it against other techniques, with results indicating superior performance. The evaluation metric used was the normalized mean square error (NMSE) across 200 test samples, resulting in an NMSE value of 0.0012, demonstrating significant improvement over alternative algorithms.

In their work [6], the authors introduced a method for SNR estimation in LTE and 5G systems. They employed a CNN-LSTM neural network, combining a convolutional neural network (CNN) with a long short-term memory (LSTM) network. The CNN was responsible for capturing spatial features, while the LSTM focused on extracting temporal characteristics from the input signal. The authors generated data using MATLAB LTE and 5G toolboxes, considering various modulation types, path delays, and Doppler shifts. Evaluation was conducted using the normalized mean square error (NMSE) metric. Remarkably, the NMSE achieved a value of zero in the time-domain across SNR levels ranging from -4 to 32 dB, indicating minimal latency. However, in the frequency-domain, the proposed method exhibited relatively poorer performance.

In their paper [7], the authors introduced NDR-Net, a novel neural network designed for channel estimation under conditions of unknown noise levels. NDR-Net consists of three main components: a noise level estimation subnet, a DnCNN, and a residual learning cascade. Initially, the noise level estimation subnet determines the noise interval, followed by processing of the pure noise image using the DnCNN. Subsequently, residual learning is applied to extract the noiseless channel image. Model performance was evaluated using the mean square error (MSE) metric across various channel models tapped delay line (TDL-A, TDL-B, TDL-C), consistently yielding low MSE values. However, it is important to note that the model's performance evaluation was limited to a SNR range of 0 to 35, which may not provide a comprehensive assessment of its robustness, especially in scenarios with high noise levels.

Finally, in [8] the authors proposed classifier based CNN for recognition of the spectrograms extracted from speech signals insert with different types of noise. Six types of noise were used to corrupt signals. They generated $30,000$ samples of data for the experiments. The spectrograms were extracted using the short time Fourier transform (STFT). Two architectures of CNN were used in the experiments, with two and three convolutional layers. The metrics indicated high performance in the proposed method.

III. Methodology

A. Proposed system

The proposed method consists of estimating the noise power density of spectrograms extracted from wireless signals using deep learning. For this purpose, the methodology is divided into four steps: (1) Signal generation, where wireless signals are generated by inserting different levels of noise power density and variables that influence the signal quality; (2) Extraction of the spectrograms, which represent the signal in the frequency domain as images; (3) Training state-of-the-art computer vision architectures; and (4) Evaluating the proposed method. Fig. 1 presents the block diagram illustrating all the described steps.

1) Signal generation: For the purposes of this paper, we will only consider hypotheses involving the presence of the primary user using the channel. We assume that a number of secondary users and a single primary user are moving at a speed v, with their starting positions randomly chosen within a given area. As a result, the users' locations change over a time interval of Δt. Additionally, we are considering a multi-channel system with N_B bands, each having a bandwidth of B_W. Furthermore, we assume that the primary user can utilize N_{B_P} consecutive bands [9]. Therefore, the received signal of the i-th secondary user on the j-th band at time n can be described as

$$y_k^j(n) = \begin{cases} s_k^j(n) + w_k^j(n), & \text{for } H_1 \text{ and } j \in B_P \\ \sqrt{\eta} s_k^j(n) + w_k^j(n), & \text{for } H_1 \text{ and } j \in B_A \end{cases} \quad (1)$$

where $s_k^j(n) = \kappa_k(n) g_k^j(n) x(n)$ and $w_k^j(n)$ is the AWGN whose noise power density is N_0, zero mean and standard deviation $\sigma_n = \sqrt{B_W 10^{\frac{N_0}{10}}}$. Being η the proportion of power leaked to adjacent bands, then B_P are the bands occupied by the primary user and B_A are the bands affected by the leaked power of the primary user.

In the expression $s_k^j(n)$, a simplified path loss model is utilized, which can be written as follows:

$$\kappa_k(n) = \sqrt{\frac{P}{\beta (d_k(n))^{\alpha} 10^{\frac{h_k(n)}{10}}}} \quad (2)$$

where α and β denote the path-loss exponent and path-loss constant, respectively. Here, $d_k(n)$ represents the Euclidean distance between the primary user and secondary user k at time n. The shadow fading of the channel, indicated by $h_k(n)$, between the primary user and secondary user k at time n in decibels (dB) can be described by a normal distribution with a zero mean and a variance of σ_s^2. The term P denotes the power transmitted by the primary user within a specified frequency band. Furthermore, the multipath fading factor, denoted as $g_k^j(n)$, is modeled as an independent zero mean circularly symmetric complex Gaussian (CSCG) random variable. Moreover, the data transmitted at time n, represented by $x(n)$, has an expected value of one [9], [10].

A special type of filter that shifts the phases of a signal while leaving all the amplitudes of the spectral components unchanged is the Hilbert transform [11].

$$\mathcal{H}\left\{y_k^j(n)\right\} = \frac{1}{\pi} \int_{m=-\infty}^{\infty} \frac{y(m)}{n-m} dm \quad (3)$$

We applied the Hilbert transform to better highlight singular information from the signals. Sequentially, we modulated the signals at a frequency ω:

$$a(n) = \left| \mathcal{H}\left\{y_k^j(n)\right\} e^{i2\pi\omega n} \right| \quad (4)$$

where $a(n)$ is the output of the signal generation step.

2) Spectrogram extractor: The Python function *scipy.signal.spectrogram* was used to extract the spectrogram from the signals, which returns a visual representation of the spectral content of the signal over time. As inputs to this function, the input signal $a(n)$ and the frequency ω are provided. Mathematically, the spectrogram is calculated by the application of the STFT, defined as

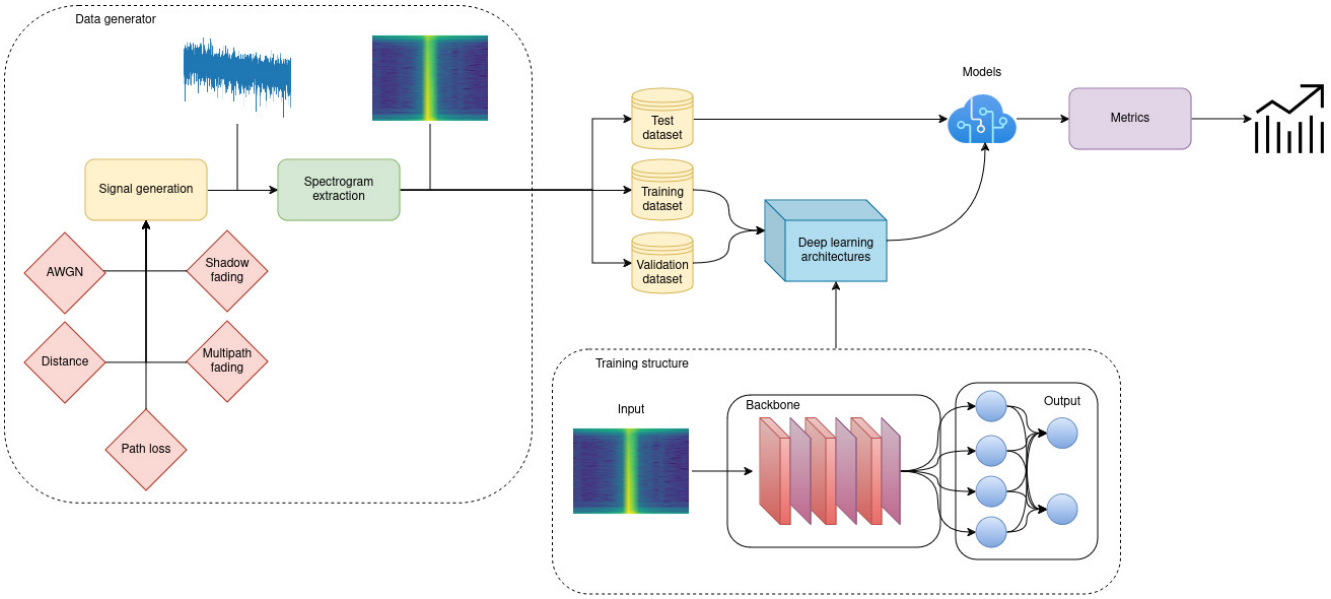


Fig. 1: Block diagram of the proposed methodology.

$$X(m,\omega) = \sum_{n=0}^{N-1} a(n) \cdot w(n-m) \cdot e^{-i\omega n} \tag{5}$$

where $X(m,\omega)$ is the spectrum of the window m in the frequency ω, $a(n)$ is the input signal, $w(n)$ the window function, and $e^{-j\omega n}$ is the complex exponential function. Then, the spectrogram is computed by the square module of the spectrum, $|X(m,\omega)|^2$. The output of the *scipy.signal.spectrogram* function consists of the window start times, frequencies in hertz, and the spectrograms calculated for each window, represented as a three-dimensional matrix.

3) Models proposed: Among the backbones deep learning models used for the noise power density estimating task are MobileNetV2, MobileNetV3Small, MobileNetV3Large, ResNet50, ResNet101, ResNet152, ConvNeXtTiny and ConvNeXtSmall. The MobileNets architecture are compose by bottleneck and convolutional layers with hard-swish (HS) as activation function

$$HS(x) = x\frac{ReLU(x+3)}{6} \tag{6}$$

where $ReLU$ is the Rectified Linear Unit. By employing this architecture, MobileNet achieves impressive performance with significantly fewer parameters compared to traditional CNNs [12].

The ResNet architecture are compose by residual units. A residual unit is designed to address the vanishing gradient problem and enable the training of very deep neural networks. The residual unit can be described as

$$y = W_2 * ReLU(W_1 * x) + x \tag{7}$$

where y is the output of the residual unit, x is the input of the residual unit and W_1 and W_2 are the weights from two convolutional layers. A residual unit applies two convolutional layers to the input, and then adds the original input to the result of these layers, enabling the network to learn identity mappings more easily and alleviating the vanishing gradient problem [13].

The ConvNeXt is a recent variant of the ConvNet architecture that apply some principles from vision transformers maintaining the convolutional nature of the model. The ConvNeXt block can be described as

$$c = GELU(W_2 * (LN(W_1 * x)) + x \tag{8}$$

where c is the output of the ConvNeXt block, $GELU$ is the Gaussian Error Linear Unit activation function and LN is the layer normalization. ConvNeXt block applies a depthwise convolution followed by layer normalization, pointwise convolution, $GELU$ activation, and then adds the original input to the result, forming a residual connection [14].

4) Metrics: The metrics used for training and testing are the categorical crossentropy (CC) and accuracy (Ac), respectively, and are described as

$$CC = -\frac{1}{N}\sum_{q=1}^{N}\sum_{z=1}^{M} y_{qz} log(p_{qz}) \tag{9}$$

where N is the number of samples, M the number of classes, y_{qz} is 1 if the sample q belongs to class z and 0 otherwise, and p_{qz} is the probability of the sample q belongs to class z.

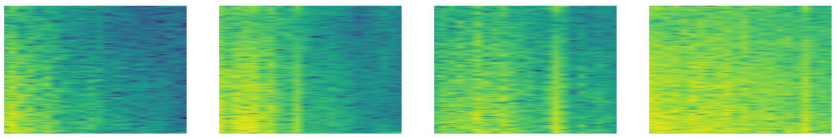
$$Ac = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

where TP are the true positives, TN are the true negatives, FP are the false positives, and FN are the false negatives.

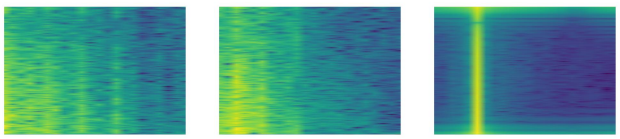
IV. Experiments and Results

A. Dataset generation

The first stage of the experiments involves signal generation. It is assumed that multiple secondary users and a single



(a) $N_0 = -114$ dBm/Hz (b) $N_0 = -124$ dBm/Hz (c) $N_0 = -134$ dBm/Hz (d) $N_0 = -144$ dBm/Hz



(e) $N_0 = -154$ dBm/Hz (f) $N_0 = -164$ dBm/Hz (g) $N_0 = -174$ dBm/Hz

Fig. 2: Examples of spectrograms extracted from signals with different N_0.

primary user are moving at a velocity of $v = 3$ km/h, and their initial positions are randomly chosen within an area of 250 meters $\times$ 250 meters. As a result, the users' positions change over a time period of $\Delta t = 5$ seconds. Each occupied band has bandwidth B_W of 10 MHz, and the primary user can simultaneously use 1 to 3 bands. Additionally, $P = 23$ dBm, $\beta = 10^{3.453}$, $\alpha = 3.8$, $\sigma = 7.9$ dB, and N_0 is randomly chosen between -114 and -174 dBm/Hz. The ratio of leaked power to adjacent bands, η, is 10 dBm, resulting in leaked power to adjacent bands being half of the primary user signal power. The carrier frequency ω used is 2.412 GHz, which is widely employed in various wireless communication standards, including Wi-Fi and Bluetooth. Signals were created with $1,024$ samples per second. For the experiments, $42,000$ instances were generated, divided into 80% for training, 10% for validation and 10% for testing. In Fig. 2 the difference between the spectrograms extracted from signals with different N_0 is shown. It is worth noting that the spectrogram is also impacted by other variables that influence the quality of the signal.

B. Training parameters

In Table I, all parameters used to train the proposed computer vision models for estimating noise based on spectrograms are presented.

TABLE I: Training parameters.

Parameter	Value
Input size	(224, 224, 3)
Layer trainable	True
Backbone	MobileNetV2, MobileNetV3Small, MobileNetV3Large, ResNet50, ResNet101, ResNet152, ConvNeXtTiny and ConvNeXtSmall
Global Average Pooling 2D	True
Dense	128
Dense	7
Epochs	$1,000$
Early stopping	True
Patience	15
Checkpoint callback	True
Optimizer	Adam
Learning rate	0.0001

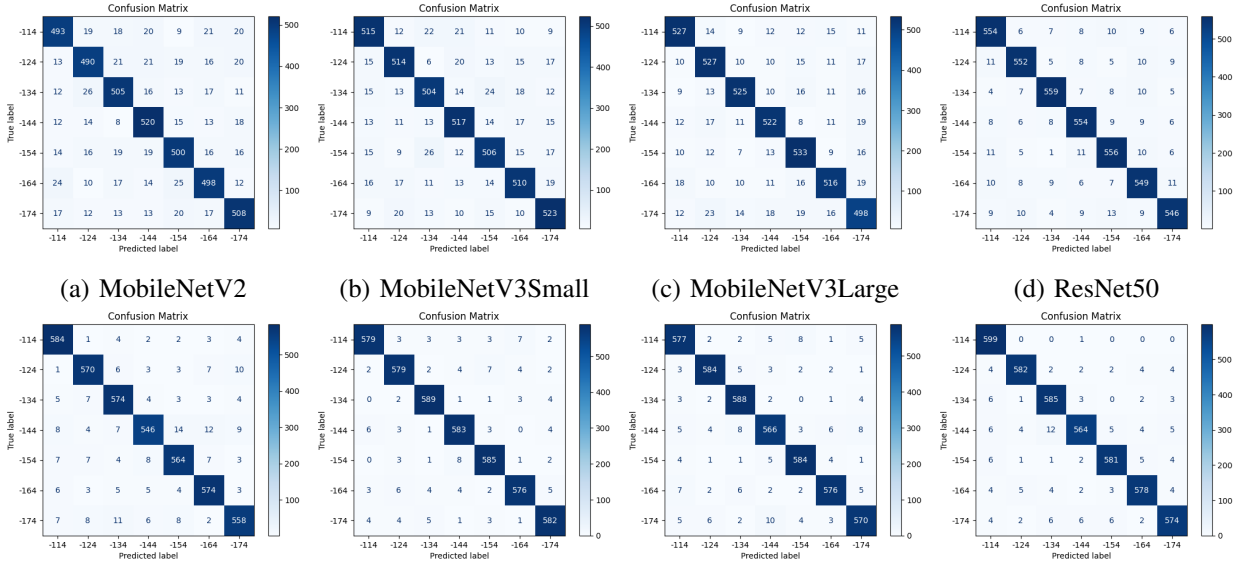
TABLE II: Accuracy achieved by each backbone on the test dataset.

Backbone	Accuracy	Training parameters
MobileNetV2	83.67%	2,422,855
MobileNetV3Small	85.23%	1,013,879
MobileNetV3Large	86.91%	3,120,263
ResNet50	92.18%	23,850,887
ResNet101	94.78%	42,921,351
ResNet152	97.05%	58,634,119
ConvNeXtTiny	96.34%	27,919,463
ConvNeXtSmall	96.72%	49,554,023

C. Noise power density estimation

In Table II, the accuracy achieved by each backbone on the test dataset is presented. Notice that the optimized architectures, MobileNets, achieved the lowest accuracy. MobileNetV2 achieved the lowest accuracy on the test dataset. However, MobileNetV3Small, which has fewer training parameters, performed better. Among the MobileNets, MobileNetV3Large achieved the best performance. The ResNets and ConvNeXts performed similarly. The networks with more parameters, in these cases, obtained the best performance, especially ResNet152, which achieved the highest level of accuracy.

In Fig.3, all confusion matrices for the backbone models proposed for the experiments are presented. All models achieved good results, but we can notice that architectures with fewer training parameters demonstrated the lowest performance. The architectures with more training parameters performed better. For example, compare Fig.3(a) and Fig. 3(f), which show the worst and best performances, respectively. We can also notice that there is no consistent pattern for class confusion; there is homogeneity in the misclassification. Furthermore, even though ConvNeXtTiny and ConvNeXtSmall have a similar number of parameters to ResNet50 and ResNet101, the ConvNeXts achieved better results. Additionally, the ConvNeXt models achieved similar results to each other.



(a) MobileNetV2 (b) MobileNetV3Small (c) MobileNetV3Large (d) ResNet50

(e) ResNet101 (f) ResNet152 (g) ConvNeXtTiny (h) ConvNeXtSmall

Fig. 3: Confusion matrices for all backbone models proposed for the experiments.

V. Conclusions

In this article, we propose the use of spectrograms extracted from wireless signals to estimate noise power density using computer vision architectures. The signals were generated based on several variables that impact their quality. The spectrograms were extracted and used as input to popular state-of-the-art computer vision architectures. The proposed method achieved a high level of accuracy, especially in deep architectures such as ResNet152, which achieved more than 97% accuracy. In this way, the proposed method proved capable of successfully estimating the noise level present in wireless telecommunication network signals, contributing to the automation process of such networks and aiding in mitigating the distortion effects of the signals received by users.

Aknowledgements

This work was a partnership between the Federal University of Amazonas, the Federal University of Uberlândia and SiDi.

References

[1] Muhammad Ali Umair, Marco Meucci, and Jacopo Catani. Strong noise rejection in vlc links under realistic conditions through a real-time sdr front-end. *Sensors*, 23(3):1594, 2023.

[2] D Smitha Gayathri and KR Usha Rani. Adapting the effect of impulse noise in broadband powerline communication. In *Proceedings of the International Conference on Cognitive and Intelligent Computing: ICCIC 2021, Volume 1*, pages 543–552. Springer, 2022.

[3] Shaoqing Zhou, Wei Xu, Kezhi Wang, Marco Di Renzo, and Mohamed-Slim Alouini. Spectral and energy efficiency of irs-assisted miso communication with hardware impairments. *IEEE wireless communications letters*, 9(9):1366–1369, 2020.

[4] Xiao Chen, Weichao Lyu, Zejun Zhang, Jian Zhao, and Jing Xu. 56-m/3.31-gbps underwater wireless optical communication employing nyquist single carrier frequency domain equalization with noise prediction. *Optics Express*, 28(16):23784–23795, 2020.

[5] Guohua Yao and Zhuhua Hu. Snr estimation method based on srs and dinet. In *Proceedings of the 2023 15th International Conference on Computer Modeling and Simulation*, pages 218–224, 2023.

[6] Thinh Ngo, Brian Kelley, and Paul Rad. Deep learning based prediction of signal-to-noise ratio (snr) for lte and 5g systems. In *2020 8th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pages 1–6. IEEE, 2020.

[7] Yinying Li, Xin Bian, and Mingqi Li. Denoising generalization performance of channel estimation in multipath time-varying ofdm systems. *Sensors*, 23(6):3102, 2023.

[8] Khalid Zaman, Cem Direkoğlu, et al. Classification of harmful noise signals for hearing aid applications using spectrogram images and convolutional neural networks. In *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pages 1–9. IEEE, 2020.

[9] Myke DM Valadão, Diego Amoedo, André Costa, Celso Carvalho, and Waldir Sabino. Deep cooperative spectrum sensing based on residual neural network using feature extraction and random forest classifier. *Sensors*, 21(21):7146, 2021.

[10] Woongsup Lee, Minhoe Kim, and Dong-Ho Cho. Deep cooperative sensing: Cooperative spectrum sensing based on convolutional neural networks. *IEEE Transactions on Vehicular Technology*, 68(3):3005–3009, 2019.

[11] Michael Feldman. Hilbert transform in vibration analysis. *Mechanical systems and signal processing*, 25(3):735–802, 2011.

[12] Myke Valadão, Lucas Silva, Matheus Serrão, Willian Guerreiro, Vitor Furtado, Natalia Freire, Gelson Monteiro, and Carlos Craveiro. Mobilenetv3-based automatic modulation recognition for low-latency spectrum sensing. In *2023 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–5. IEEE, 2023.

[13] Wanni Xu, You-Lei Fu, and Dongmei Zhu. Resnet and its application to medical image processing: Research progress and challenges. *Computer Methods and Programs in Biomedicine*, page 107660, 2023.

[14] Hongbin Zhang, Xiang Zhong, Guangli Li, Wei Liu, Jiawei Liu, Donghong Ji, Xiong Li, and Jianguo Wu. Bcu-net: Bridging convnext and u-net for medical image segmentation. *Computers in Biology and Medicine*, 159:106960, 2023.