Vessel Classification through Convolutional Neural Networks using Passive Sonar Spectrogram Images

Lucas P. Cinelli, Gabriel S. Chaves, Markus V. S. Lima

***Abstract*— Vessel classification is an extremely important task for coastal areas security and surveillance. Currently, this task relies on Synthetic Aperture Radar (SAR) images but gathering these images is expensive and often prohibitive. In this paper, we propose using spectrograms containing characteristic sound noise records of each vessel acquired from a single passive sonar device as an input to a convolutional neural network, which performs the classification. The main advantage of our method is its simplicity and low cost development due to the nature of this kind of data. Furthermore, our proposal can be used alongside other SAR-image-based method, potentially improving results of the overall classifier.**

***Keywords*— convolutional neural networks, classification, machine learning, security, surveillance, image processing, passive sonar, spectrograms**

I. Introduction

Vessel classification is an essential task for coastal areas security and surveillance, being indispensable for the military forces, in particular, the Marine. A device capable of identifying the class or type of a ship would assist on a wide range of situations, such as nation coastal defense, war, or simply keeping records of a less monitored channel, as in [1].

Classification of marine vessels is not a recent problem, and different techniques were developed over the years to address it. However, most of them rely on Synthetic Aperture Radar (SAR) images combined either with classical machine learning algorithms [2], [3] or with deep neural networks techniques in more recent approaches [1], [4]–[6].

Recently, the popularization of convolutional neural networks (CNN) set a new level on the state-of-the-art results not only in image processing tasks, for which it was originally developed [7], [8], but also in audio [9], voice [10] and others. The CNN structure is loosely based on the human visual cortex, that allows hierarchical extraction of images attributes [11], from simple low level features such as border and edge detection to complex abstractions like gender and object type.

Naturally, there is a growing trend over the last years of applying CNN to vessel detection and classification [1], [4]–[6]. However, those methods rely on SAR images, which are difficult to acquire, thus imposing a high cost. The present work proposes spectrograms obtained from passive sonar recordings, which contains characteristic sound noises of the ships, as input to the network model, instead of the expensive SAR images. These recordings make the development and implementation costs cheaper, since they are easily obtained through hydrophone measurements.

Spectrogram usage is very common in many other areas. For example, numerous applications in audio processing use the spectrogram on convolutional neural networks [11], [12]. These works aim to identify characteristic patterns from a determined sound source, essentially the same problem faced in ship classification. It is worthwhile to emphasize that, like musical instruments, each vessel has its own characteristic sound, its signature.

This paper's outline is as follows. In Section II, general concepts about neural network, CNN architectures and the motivation behind this type of network are presented. In Section III, we explain how the database was originally generated. In Section IV, we address the experimental procedure, the baseline models, and the different CNN configurations employed. In Section V, we present and discuss the results and, for last, the conclusions are drawn in Section VI.

II. Neural Networks

The atomic unit in a neural network is the neuron. This structure computes the weighted sum of its inputs and applies a nonlinear function on the result, such non-linearity varies according to the application at hand. A set of non-interconnected neurons defines a layer, hence they operate independently and simultaneously [13].

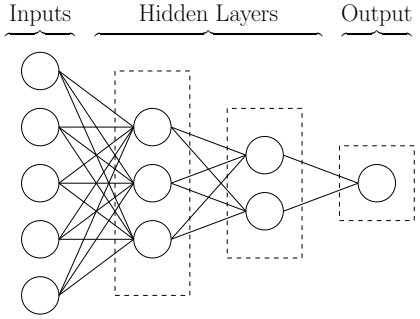


Fig. 1: Simple neural network diagram.

Figure 1 illustrates a simple neural network. This figure shows a dense network, where all the neurons in the same layer are fully connected with those in the previous layer. One can notice that the network depicted has 5 inputs and 3 layers, where the middle ones are called hidden layers and the last one is the output. The number of layers determines the network depth [14].

Neural networks have the capacity to approximate any continuous function, depending on the chosen parameters [15],

Lucas P. Cinelli, Gabriel S. Chaves, Markus V. S. Lima, Federal University of Rio de Janeiro (UFRJ) / Polytechnic School (Poli) / Program of Electrical Engineering (PEE) / Signal Multimedia and Telecommunications (SMT), Rio de Janeiro - RJ, Brazil, E-mails: { lucas.cinelli, gabriel.chaves, markus.lima}@smt.ufrj.br.

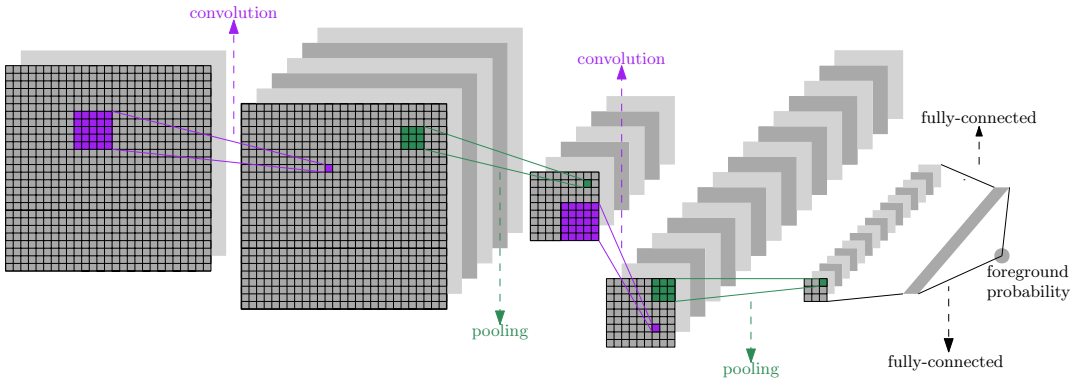


Fig. 2: Common schematic for a CNN.

through a process called *learning*. This is an advantage when using this class of algorithm: the capacity of learning functions that represent specific types of data (used for training the network) with a given precision [15].

A. Convolutional Neural Networks (CNNs)

Convolutional neural networks are so called because they rely on convolutional layers to extract features from the input. This type of network is often used in image processing, since CNNs assume their inputs are graphical representations of some kind and build into the architecture a priori knowledge, such as translation invariance. Hence, they are capable of extracting patterns more easily. Nonetheless, CNNs already found widespread use with many others input types [12], [16]. Convolutional networks are generally combined with standard neural networks; while the former extracts the features of interest, the latter, generally comprised of one or more layers at the end of the network, is responsible for classifying the data [17].

A convolutional layer consists of several independent filters that operate locally on its inputs. Each filter kernel slides through the whole input with the desired stride computing the inner product with the overlapping region at each given position, that is, it implements a 2D convolution. The output positions and values compose an activation map, from which the most relevant regions may be extracted, for each filter, and fed as input to the next layer, possibly convolutional [17].

The other cornerstone of CNNs is the pooling layer, responsible for locally aggregating information. They consist of filters that operate independently on each channel (feature map), differently from filters in convolutional layers, which operate simultaneously on all channels. Pooling layers shrink the image dimension through a predetermined statistic criterion by replacing the information present in an individual pool by a single value, most commonly its maximum or average. This procedure renders the model less complex [17] and progressively selects the most relevant features.

A typical convolutional neural network, with all the layers presented so far, is shown in Figure 2.

III. Database

The database comprises 263 runs[1] divided in 8 different classes (A, B, C, D, E, F, G, H). Each class has among 2 and 5 distinct vessels and number of runs varying from 19 to 66. Each run was discretized with a sampling rate of $F_s = 22.05$ kHz and 16 quantization bits [18].

Acoustic data of the underwater channel were recorded by a submerse hydrophone while a single vessel traversed the acoustic ray. The measurement conditions were the kept as controlled and constant as possible during the experiment. For more details refer to [18].

From the audio signals, their 4096-point Fourier Transforms were computed using a non-overlapping Hanning window of the same size and from the 4096 output bins, only the magnitude of the first 557 bins were considered, which corresponds to a $0 - 3$ kHz range in the original frequency domain. This range was chosen because it contains relevant vessel-specific information, whereas higher frequencies are related to more general features [18]. Next, a Two Pass Split Window (TPSW) algorithm estimates the background noise, which is then removed [19]. Finally, the power spectrum of the resulting signal is normalized so that its energy sums to unity [18]. At the end, there are 263 matrices of size $L \times 557$, which are the spectrograms for all runs, and where the length L depends on how many windows fit the original audio. A block diagram representing the preprocessing steps is depicted in Figure 3.

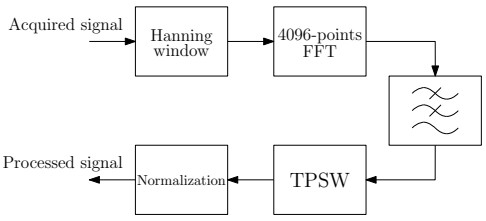
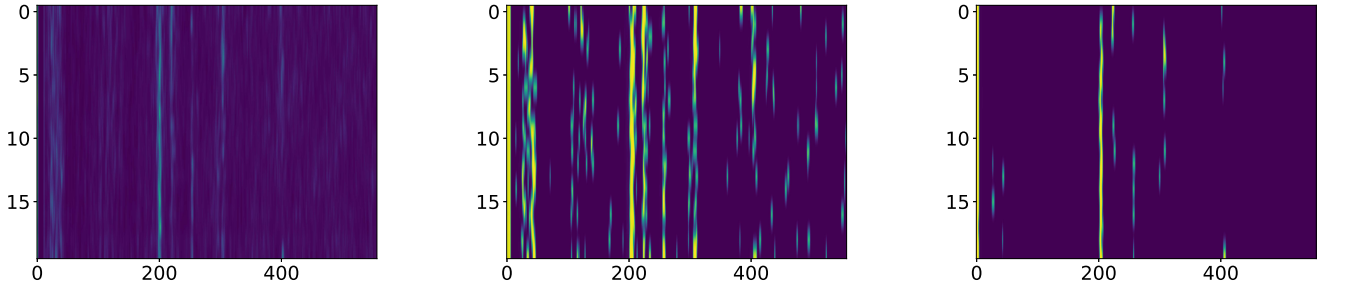


Fig. 3: Block diagram of the preprocessing steps.

It is worth mentioning that the database is highly unbalanced both in number and duration of the runs through the classes. Thus, possibly leading to a classifier heavily biased towards the dominant classes. This issue is addressed, as described in

[1]A run is the acoustic noise recording of a ship navigating through a predefined route, called acoustic ray, maintaining the same operation condition throughout the whole course.



(a) CNN channel 1 input.

(b) CNN channel 2 input, delta frequency

(c) CNN channel 3 input, delta-delta frequency.

Fig. 4: Network channel inputs.

Section IV, so as to guarantee equally probable class sampling during training.

IV. Experimental Procedure

As detailed in Section III, the database consists of 263 spectrograms with 557 frequency bins each. Their vertical axes carry temporal information about the run while the horizontal axes, frequency information about the underwater channel during the vessel transit. However, we do not input all temporal bins of a run to the network at once. Instead, we split it in as many H-length spectrograms as possible, by sampling with an H-length temporal window either densely (in training) or without overlap (in test), thus resulting in several different $H \times 557$ matrices.

As baseline models for our experiment, we employ a simple multilayer perceptron (MLP) with no hidden layers and another with a 512-neuron hidden layer, hereafter named MLP0 and MLP1, respectively. Their inputs are the flattened (1D) windowed matrices of length 21, which approximately corresponds to a 4-second time frame. The length value choice is related to the duration of the briefest runs. We set the learning rate to 1×10^{-2}, and weight decay to 3×10^{-3}.

The convolutional networks designed use as the final classifier an MLP with a *softmax*, a widely used function for multiclass problems. The loss function minimized by the network during the learning phase is

$$H(p, q) = -\sum_{x} p(x) \log q(x), \tag{1}$$

and is known as cross-entropy, where $p(x)$ is the probability density function over the labels and $q(x)$ its estimation. The cross-entropy is commonly used in classification problems due to its convergence properties over the mean squared error cost function. The optimizer we use is the ADaptive Momentum (ADAM), a first order gradient descent algorithm that adaptively tune the learning rates for each parameter during training [20]. ADAM is one of, if not, the most frequently used optimizer on CNNs. Learning rate and weight decay are set to 3.85×10^{-3} and 8.9×10^{-3}, respectively. Moreover, we train all models for 70 epochs and divide the learning rate by 10 every 30 epochs. It is worth mentioning that even though hyperparameters may be the same, results of different simulations are never exactly equal because sampling and weights' initialization are both stochastic.

We split the 263 runs of the database into 3 disjoint sets: training, validation, and test, following the proportion 70%, 15%, and 15%, respectively. In order to address the database unbalance issue, sampling is performed in a stratified random fashion which aims to render all classes equiprobable. After a full run is selected, a window sample of length L is chosen. The window position is drawn from a uniform distribution. This approach guarantees independence between the 3 sets. Sample manual separation is impracticable, because it is hard to correctly evaluate each run in a way to create representative sets for validation and test.

We have evaluated the use of up to three input channels, of which the spectrogram is just the first of them. The other two are delta frequency images, and delta-delta frequency images [21]. These data are the results of operations on the base spectrum, and, essentially, correspond to different-order derivatives. Since spectrograms exhibit, for the most part, a static view of the system, these new features somewhat capture the dynamics of it. An example of all 3 channels for the same original recording (class E, ship 1) is depicted in Figure 4.

The sequence of operations of the proposed CNN architecture, hereafter called VesselNet, is detailed below:

1) 2d convolution: 32 filters with size 4×512;
2) Batch Normalization;
3) ReLU;
4) Max Pooling: 3×4 kernel with stride of 3×2;
5) 2d convolution: 32 filters with size 3×2;
6) Batch Normalization;
7) ReLU;
8) Max Pooling: 2×3 kernel with stride of 1×3;
9) Fully connected: 128 neurons;
10) ReLU;
11) Output: 8 classification neurons;

where ReLU is the Rectified Linear Unit, the nonlinear element-wise function used [22], and Batch Normalization [23], a technique that improves the network stability by normalizing the layers' activations to zero mean and unit variance, and is widely used, specially for deeper networks. Although different from the VesselNet, the network in Figure 2 depicts how the above operations are interconnected.

One may notice that filters, particularly those in the first layer, are larger along the horizontal axis. This property results in filters less invariant to horizontal translations and, consequently, more sensitive to large frequency variations, a

fundamental aspect in distinguishing different classes.

V. Results

This section presents and discusses the results obtained by the different studied algorithms.

Firstly, we analyze whether the use of delta (and delta-delta) features improve performance. One should notice that using more channels forcefully implies in considerably increasing the number of parameters and thereby the network complexity, specially in the MLP case, which may actually degrade the performance. We achieved the best results for the MLP0 baseline model when using 2 channels (spectrogram and delta frequency), reaching 79.5% average precision on the validation set with the no-hidden-layer MLP. Nevertheless, such configuration produces scattered decisions with a reasonable amount of mistakes throughout several classes, as we observe in the confusion matrix of Figure 5. The confusion matrix is a graphical representation of the model's performance in which the horizontal axis exhibits the predicted labels and the vertical axis the true labels. In this way, good classifiers are related to diagonal dominant confusion matrices.

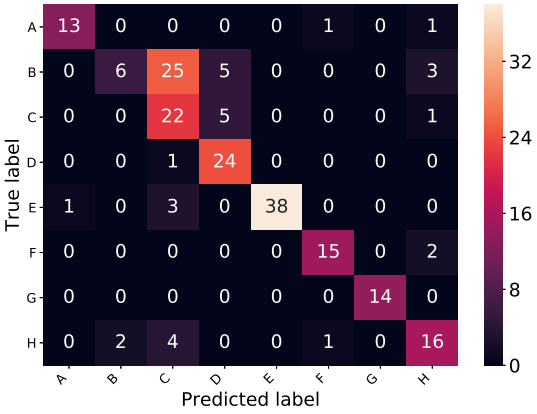


Fig. 5: MLP0 baseline confusion matrix on validation.

As for the MLP, VesselNet attained best results when using on the same 2 channels as input (spectrogram and delta frequency). It reaches 83.9% average precision on validation, a score 4.4% higher, which corresponds to an 5.5% improvement over the MLP0 baseline. The network's inference time is 10 ms for an input dimension of 21×557, that is, a 21-length temporal window. We notice from Figure 6 that the VesselNet classification is less disperse, the misclassification is more concentrated on fewer classes. For example, the MLP0 (Figure 5) misclassifies H as either B, C or F, while VesselNet (Figure 6) mistakes H only for C.

We notice from Figure 7 that both models, the MLP0 baseline and VesselNet, struggle during training (MLP1 behaves similarly). Their learning phases are unstable, though they converge at the end. The VesselNet presents large peaks in the loss function (Figure 7a) and corresponding valleys in the precision curve (Figure 7b). The batch average precision (Figure 7b) of both networks reaches nearly 100% during training, while it attains about $\sim$80% on validation, such

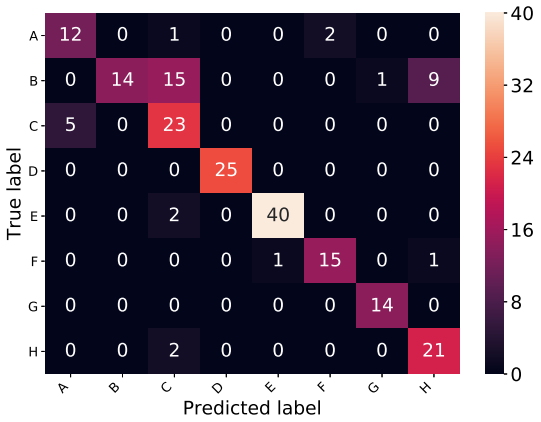


Fig. 6: VesselNet confusion matrix on validation.

behavior suggest overfitting is occurring. Typically, the main reasons for that are large number of trainable parameters and small database. Moreover, our signals are very similar among themselves. The use of the additional delta-delta channel, adding to a total of 3 channels, leads to lower variance and smaller overfitting. However, the final average precision is also $\sim$5% smaller and, thus, not considered in our analysis.

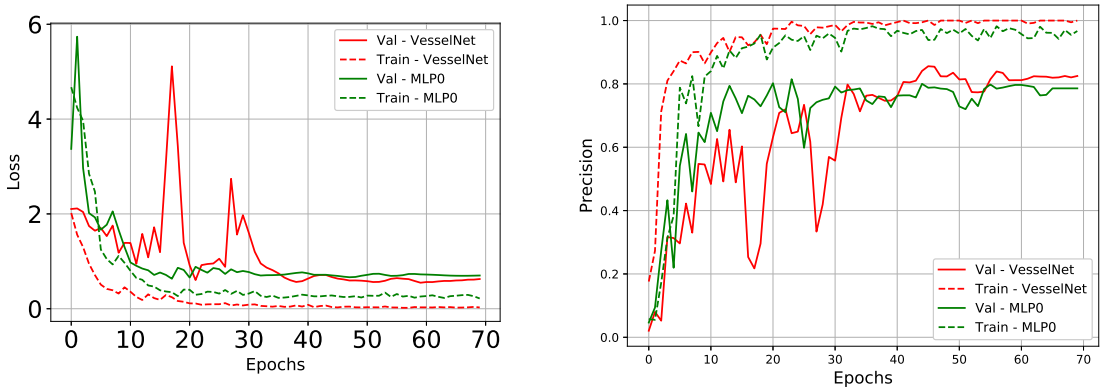
Besides having an $\sim$5% higher precision, the 2-channel VesselNet has fewer parameters than the baseline MPL0. Indeed, VesselNet uses $187,160$ parameters whereas MLP0 requires $224,584$. Although the MLP1 model indeed reaches a better average precision than its counterpart, 79.7% (a modest 0.2% better than the no hidden layer version) it has 12×10^6 parameters, about 50 times more than the others .

Finally, we evaluate the VesselNet model on the test set and achieve 88.1% average precision, 4.2% above the validation set. The confusion matrix is depicted in Figure 8. Similarly to all other models on any of the sets, class B is highly misclassified: out of its 22 samples, only 10 were correctly predicted. Hence, confirming the level of difficulty of learning this class.

VI. Conclusion

We conclude that the VesselNet attains better results using only 2 channels, the spectrogram and the delta-features, improving performance on 4.3% (average precision). Some improvement was already expected since there are not so many images from which the network may learn, thus handing higher-level features eases the task of feature extraction and bootstraps training. Furthermore, employing the additional delta-delta channel decreases average precision by $\sim$5%, though has the advantages of lower variance during training and smaller overfitting.

The VesselNet was capable of classifying and distinguishing the different ships by using spectrogram images of audio recordings of a passive underwater sonar as input. These data are easier and cheaper to obtain than current SAR-image-based techniques. It is worth mentioning that the networks suffered overfitting, and the main reasons for that are: small database, large number of network parameters, absence of an



(a) Learning curves for cost function.

(b) Learning curves for precision.

Fig. 7: VesselNet and baseline learning curves.

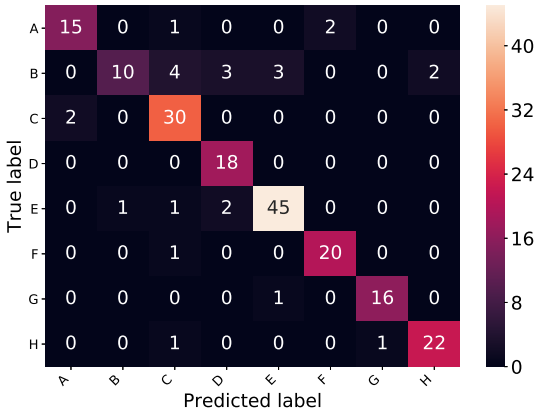


Fig. 8: VesselNet confusion matrix on test.

effective method to enlarge the database, as well as high inter-class similarity. While there are some techniques to avoid the overfitting, such as data augmentation, the effective ones require information about the background noise, which we do not have.

VII. Acknowledgements

The authors would like to thank CAPES, CNPq, and FAPERJ agencies for funding this research work.

References

[1] N. Odegaard, A. O. Knapskog, C. Cochin, and J.-C. Louvigne, "Classification of Ships using Real and Simulated Data in a Convolutional Neural Network," in *2016 IEEE Radar Conference (RadarConf)*. Philadelphia, United States: IEEE, may 2016, pp. 1–6.

[2] G. Margarit and A. Tabasco, "Ship Classification in Single-Pol SAR Images Based on Fuzzy Logic," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 8, pp. 3129–3138, aug 2011.

[3] R. G. V. Meyer, W. Kleynhans, and C. P. Schwegmann, "Small Ships don't Shine: Classification of Ocean Vessels from Low Resolution, Large Swath Area SAR Acquisitions," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 2016-Novem. Beijing, China: IEEE, 2016, pp. 975–978.

[4] C. Bentes, A. Frost, D. Velotto, and B. Tings, "Ship-Iceberg Discrimination with Convolutional Neural Networks in High Resolution SAR Images," *11th European Conference on Synthetic Aperture Radar Electronic Proceedings*, pp. 491–494, 2016.

[5] C. Bentes, D. Velotto, and S. Lehner, "Target Classification in Oceanographic SAR Images with Deep Neural Networks: Architecture and Initial Results," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 2015-Novem. Milan, Italy: IEEE, jul 2015, pp. 3703–3706.

[6] M. Kang, X. Leng, Z. Lin, and K. Ji, "A Modified Faster R-CNN based on CFAR Algorithm for SAR Ship Detection," in *2017 International Workshop on Remote Sensing with Intelligent Processing (RSIP)*. Shangai, China: IEEE, 2017, pp. 1–4.

[7] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "High-Performance Neural Networks for Visual Object Classification," *Biochemical and Biophysical Research Communications*, vol. 330, no. 4, pp. 1299–1305, 2011.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks."

[9] Y. M. Costa, L. S. Oliveira, and C. N. Silla, "An Evaluation of Convolutional Neural Networks for Music Classification using Spectrograms," *Applied Soft Computing*, vol. 52, pp. 28–38, mar 2017.

[10] I. Macedo Quintanilha, "End-to-end Speech Recognition Applied to Brazilian Portuguese Using Deep Learning," Ph.D. dissertation, Federal University of Rio de Janeiro, 2017.

[11] K. Choi, G. Fazekas, and M. Sandler, "Automatic Tagging Using Deep Convolutional Neural Networks," *2016 Conference International Society of Music Information Retrieval (ISMIR)*, pp. 1–7, 2016.

[12] M. Dorfler, R. Bammer, and T. Grill, "Inside the Spectrogram: Convolutional Neural Networks in Audio Processing," in *2017 International Conference on Sampling Theory and Applications (SampTA)*, no. 1. Tallin, Estonia: IEEE, jul 2017, pp. 152–155.

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, 1st ed. Cambridge, United States: MIT Press, 2016.

[14] Y. Bengio, "Learning Deep Architectures for AI," *Foundations and Trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[15] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, "Multilayer Feedforward Networks with a Nonpolynomial Activation Function can Approximate any Function," *Neural Networks*, vol. 6, no. 6, pp. 861–867, jan 1993.

[16] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep Reinforcement Learning: A Brief Survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, nov 2017.

[17] R. d. M. E. Filho, "A Study on Deep Convolutional Neural Networks for Computer Vision Applications," Ph.D. dissertation, Federal University of Rio de Janeiro, 2017.

[18] J. B. d. O. S. Filho, "Classificação Neural de Sinais de Sonar Passivo," Ph.D. dissertation, Federal University of Rio de Janeiro, 2007.

[19] R. O. Nielsen, *Sonar Signal Processing*, 1st ed. Norwood, United States: Artech House, Inc., 1991.

[20] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," pp. 1–15, 2014.

[21] K. J. Piczak, "Environmental Sound Classification with Convolutional Neural Networks," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, vol. 2015-Novem. Boston, United States: IEEE, 2015, pp. 1–6.

[22] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," *Proceedings of the 27th International Conference on Machine Learning*, jun 2010.

[23] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proceedings of The 32nd International Conference on Machine Learning*, Lille, France, jul 2015.