Loss Function Design For Training Robust Radar Detectors Using Deep Learning

Noé Lallouet
LAMSADE, Université Paris-Dauphine
Thales DMS
Paris, France
noe.lallouet@thalesgroup.com

Tristan Cazenave
LAMSADE, Université Paris-Dauphine
Paris, France
tristan.cazenave@lamsade.dauphine.fr

Cyrille Enderli
Thales DMS
Elancourt, France
cyrille-jean.enderli@fr.thalesgroup.com

Stéphanie Gourdin
Thales DMS
Elancourt, France
stephanie.gourdin@fr.thalesgroup.com

***Abstract*—Recent advances in research show interesting potential in using deep neural networks to perform the task of radar target detection. For radar applications, especially in the military domain, the detection method is designed to follow the Neyman-Pearson criterion with the aim to maximise the detection probability while keeping the false detection rate controlled. While standard CFAR (Constant False Alarm Rate) detection is designed to fit this need, it is not the case of neural networks that do not naturally prioritize detection over false alarm rate. In this paper, we propose an overview and a comparison of different loss functions, namely Tversky Loss and a loss based on constrained optimization, for training deep CNNs on the problem of radar target detection, with the objective to get a better compromise between false alarm rate and detection. We then demonstrate that the models obtained with these methods outperform the baseline CNN model as well as classic CFAR detectors. The developed models are compared on the detection probability (P_D) and false alarm probability (P_{FA}) criteria on exoclutter[1] environments. Model evaluation on thermal noise is an important step for validating a detector and is rarely explored in related research. It is found that training the neural network with a loss function constrained by the expected false alarm probability provides higher detection probability at a fixed P_{FA} than the baseline models. The advantages and shortcomings of training the detector with the Tversky loss function are also highlighted.**

***Index Terms*—Radar, target detection, deep learning, optimization**

I. Introduction

Target detection is one of the most fundamental problems in radar signal processing. On exoclutter environments, target detection consists in solving the following binary decision problem :

$$\begin{cases} H_0 : y(t) = \nu(t) : \text{absence of target} \\ H_1 : y(t) = x(t) + \nu(t) : \text{presence of target} \end{cases} \tag{1}$$

where $y(t)$ is the received signal, $\nu(t)$ is the thermal noise signal and $x(t)$ is the signal of the target. This paper addresses detection in the setting of a Pulse-Doppler radar. The problem of target detection is optimally solved with statistical signal processing methods such as CFAR [1], but these methods require strong assumptions about environment and target unicity that, in real situations, are not always observed.

The use of deep learning applied to radar target detection has risen in recent years, having shown promising results. However, a critical aspect that remains underexplored is the ability to build reliable detectors and provide performance guarantees. Our work contributes to filling this gap in research by investigating ways to train the model with the goal of satisfying a constraint on P_{FA}. Moreover, scarce are the research works solely focusing on deep learning-based detector performance evaluation over exoclutter environments. This crucial assessment needs to be performed in order to validate neural detectors operationally.

It is found that radar detection performance improves when training the neural network with different loss functions than the ones that have been focused on in related research.

II. Related Works

In recent years, the ever-increasing available computation power and existence of large datasets have enabled deep learning models to revolutionize the way numerous problems are approached, including natural language processing, time series forecasting, computer vision, etc. In particular, convolutional neural networks (CNNs) have been successfully used to perform different vision tasks, such as image classification ([2], [3], [4]) or segmentation ([5], [6], [7]).

The problem of radar target detection on range-Doppler maps, in the operational setting used in this paper, can be expressed as a single-point image segmentation problem. Indeed, after signal processing, the target appears as a singular point in a 2D array. As such, the use of classic image segmentation architectures applied to radar signal processing is of particular interest. After neural network inference, the post-processing chain performs non-max suppression in order to prevent

[1] Exoclutter refers to a situation where there is an absence of unwanted echoes.

eventual small inaccuracies of around one pixel in the predicted segmentation map.

Today, radar target detection is performed using Constant False Alarm Rate (CFAR) detectors [1]. These detectors estimate the variance of thermal noise in a neighbourhood of the cell under test (CUT) and announce a detection if the value of the CUT is superior to a certain threshold, which is normalized with respect to the estimation of local noise power. The threshold is computed with respect to the estimated noise variance. CFAR detectors, under strong assumptions, are statistically optimal. However, the assumptions in which CFAR detectors are optimal (absence of clutter[2] and/or side lobes, unicity of target, range resolution superior to target size...) are seldom met, resulting in suboptimal target detection performance. Even tough techniques (GO-CFAR, SO-CFAR...) [8] have been developed to mitigate this, there exists a margin to develop detectors more polyvalent than CFAR-based detectors.

Radar signal processing has also been subject to deep learning-based innovation. Deep learning has been successfully used to perform radar waveform recognition [9], automatic target recognition [10], among others. The works of [11], [12] leverage convolutional neural networks (CNNs) to detect targets in the 4D space (range, Doppler, azimuth and elevation), while [13] and [14] focus on CNN detectors on range-Doppler maps. The works of [15] implement a Faster R-CNN network ([16]) in order to automatically detect marine targets.

The development of neural networks capable of guaranteeing a fixed P_{FA} has also been the object of research. Research by [17] promotes a CFAR neural detector by training a network to minimize a statistical distance between the network output of two examples belonging in the H_0 hypothesis (Equation 1). The work of [18] leverages a CNN that detects and masks a target from a range-Doppler map in order to improve the noise estimation performed by a classic CFAR detector. Finally, [19] introduces a novel constrained loss function related to the Neyman-Pearson criterion for marine target detection. The network of [19] outperforms CFAR detector and a CNN trained using the cross-entropy loss. We propose to extend this comparison of a constrained loss against other loss functions. Furthermore, even though we aim to develop a polyvalent detector, the scope of our analysis is restrained to the exoclutter scenario, where comparing detection and false alarm performances is straightforward.

III. Methodology

A simulation model that is able to generate realistic radar echoes is used to create the data necessary for training the model. The radar signal generation tool is validated and generates data which is representative of real data. The benefits of using such a generator are threefold : first, our ability to generate data is only limited by computation and time constraints. Second, it can be chosen to produce data that represents the distribution of real-world applications. Finally, the correctness of the ground truth labels is guaranteed.

The data consists in 100000 range-Doppler maps on which have been added 5 simulated targets. The number of 5 is chosen with the aim of increasing the number of positive examples in the training dataset, thus reducing data imbalance. Varying the number of targets on a training example, by, for example, randomly choosing a number of targets during simulation, and shifting the variance of thermal noise might be of interest in future work. The range-Doppler maps are generated according to a number of scenarii, including both endoclutter and exoclutter situations. The motivation for training the model on various noise and clutter profiles resides in the will to create a model able to generalize detection on exoclutter environments to more complex ones. Each target has a distance and velocity that is randomly drawn inside the operational domain according to a bivariate uniform law. Additionally, the SNR (signal-to-noise ratio) of the generated images is tuned by varying the radar cross section (RCS) of the targets, thus effectively varying the reflected power received by the radar.

Research works such as [10] and [11] have shown that neural networks, and especially CNNs, are adapted to the task of radar target detection. In accordance with these findings, a CNN architecture is used to perform the task. As the problem of target detection on range-Doppler maps can be formulated as an image segmentation problem, a classic U-Net encoder-decoder structure [6] is leveraged. The U-Net architecture consists in several encoding stages that express the input in a lower-dimensional latent space, which are followed by decoding stages that take the input back to its original resolution. Any two pair of encoding and decoding stages of the same rank k are connected by a *shortcut connection*, that brings back the necessary spatial information. The shortcut connection is an `Add` operator taking as inputs the features map for rank k of the encoder and decoder. The range-Doppler maps are resized to the classical tensor dimensions of 256×256 and given as input to the model. The architecture of the network is detailed in Figure 1. The sigmoid activation function is applied to the final layer activation values, outputting a value between 0 and 1 representing a confidence value of classifying the pixel as background noise or as a target. The neural network has a total of $2,121,481$ parameters.

The network is trained on the aforementioned simulated range-Doppler maps. A training / validation split of 80/20% is used. The test data is generated on-the-fly, and therefore has not been used during training. It consists in a singular target in a simulated exoclutter environment. Neural network testing is performed with 20,000 generated images. The performance metrics such as detection probability P_D and false alarm probability P_{FA} are computed after the detection post-processing chain used in the radar detection pipeline. As such, these metrics are representative of detection performance in real scenarii.

Network training is performed on a Nvidia GTX 1080 Ti using the Adam optimizer and a learning rate decreasing from 7×10^{-3} to 5×10^{-5} for 150 epochs. Model values with

[2]Clutter is defined as unwanted echos received by the radar, e.g. coming from the ground. In our operational setting, ground clutter is expected to cover roughly half of the range-Doppler map.

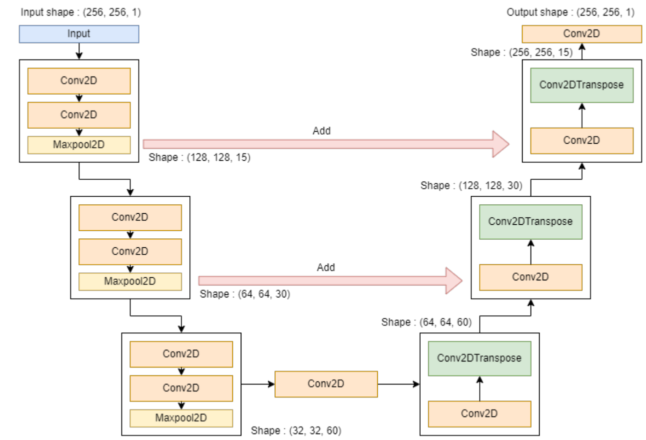

Fig. 1. Architecture of U-Net

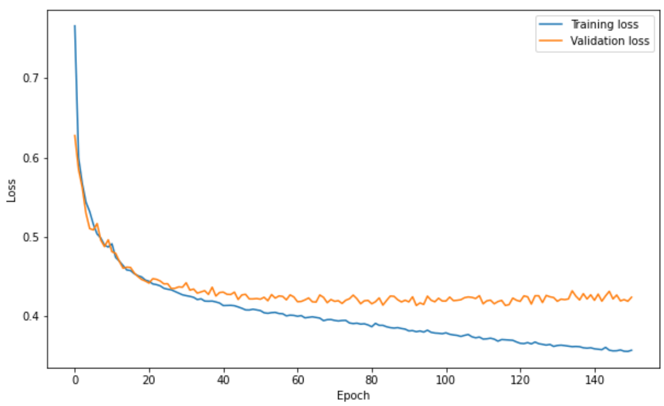

Fig. 2. Evolution of training and validation loss for the baseline CNN model

the lowest validation set loss are selected, in order to reduce overfitting. The training and validation loss evolution curves for the baseline CNN model trained with the Dice loss function is shown in Figure 2. We have not encountered any specific difficulties for model training, as weight convergence remains stable for any training run.

Loss function

A large number of research works applying CNNs to the task of radar target detection use popular loss functions such as cross-entropy, weighted cross-entropy or the Dice coefficient for training. Even though these choices are valid and lead to good detection performances, they are not designed to fit the very specific radar detection problem. As such, it remains to be seen whether training the network with a different loss function can improve P_D at a fixed P_{FA} over baseline models. It has been previously shown [20] that using the Dice coefficient (2) as a loss function during model training leads to improved segmentation accuracy over weighted cross-entropy loss in the case of imbalanced data. Due to the sparse nature of the data, where the positive-to-negative label ratio is 7.6×10^{-5}, it has been chosen to use Dice coefficient as a baseline.

$$D(\hat{y}, y) = \frac{2\sum_i^N \hat{y}_i y_i}{\sum_i^N \hat{y}_i^2 + \sum_i^N y_i^2} \tag{2}$$

where $\hat{y}_i$ and y_i are respectively predicted pixels and ground truth pixels. N is the total number of pixels in an image. $\hat{y}_i$ belongs to the interval $]0;1[$, while y_i belongs to the set $\{0, 1\}$. When using mini-batch stochastic gradient descent, which is the case of this research work, the value D is calculated as the average of the Dice coefficient for every image in a batch of size m. The CNN trained using the Dice Loss $DL = 1 - D$ as a cost function will be referred to further as **NN-Dice**.

We formulate the hypothesis that convolutional neural networks trained using the Dice coefficient as a loss function show limits in performance due to the fact that the Dice coefficient aims to maximize the intersection over union of ground truth and prediction pixels. As such, the loss function assigns the same weight to false negatives (missed detections) and false positives (false alarms). As the results presented in Section IV show, a network trained using the Dice coefficient results in a high-precision and low-recall detector. It is then of interest to train a detector that exhibits a higher recall, while staying under the required maximum number of false alarms.

We introduce the use of the Tversky Loss [21] during training of the model. Tversky Loss is an extension of Dice loss with added coefficients which enable to control the importance that is given to false positives and false negatives during training. Tverksy Loss has been investigated in automotive radar applications in [22]. Using previous notation, the Tversky Index TI can be written as follows :

$$TI = \frac{\sum_i^N \hat{y}_i y_i}{\sum_i^N \hat{y}_i y_i + \alpha \sum_i^N \hat{y}_i(1 - y_i) + (1 - \alpha)\sum_i^N (1 - \hat{y}_i) y_i}$$

The cost function that is optimized during training is $TL(\hat{y}, y) = 1 - TI(\hat{y}, y)$. It may be noticed that $\sum_i^N \hat{y}_i y_i$ represents true positives (TP), $\sum_i^N \hat{y}_i(1 - y_i)$ false positives (FP) and $\sum_i^N (1 - \hat{y}_i) y_i$ false negatives (FN). Thus, the parameter α may be used to give more or less importance to one or the other during training.

The hyperparameter α is found using a grid search and selecting the model that maximizes P_D at a fixed P_{FA}. To the best of the authors' knowledge, Tversky Loss has not been used during the training of neural networks in the context of radar target detection.

The neural network trained using TL will be referred to as **NN-Tversky**. As it shall be demonstrated in the results section, using Tversky Loss improves the detection probability of the algorithm. However, even though the false alarm rate remains inferior to the value at which the CFAR detector is calibrated, it remains unclear how to guarantee an expected false alarm rate for neural detectors.

False alarm control

Radar detection systems operate under the constraint of producing a fixed number of false alarms during a period of time. The classical value is set at 1 false alarm (FA) per

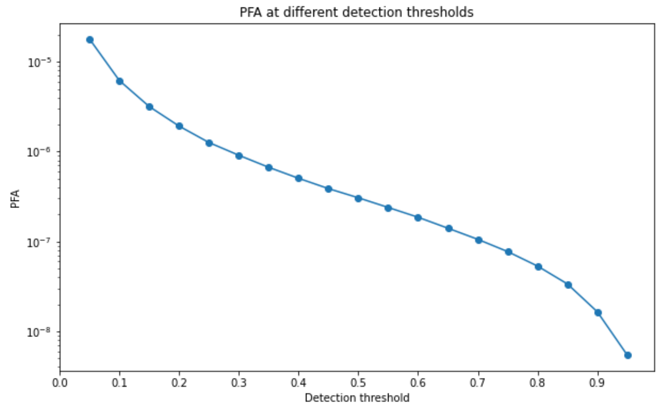

Fig. 3. P_{FA} at varying detection thresholds for the Dice loss function

minute. On exoclutter backgrounds, the dynamic threshold computation guarantees —notwithstanding CFAR loss—a statistically maximal P_D at a fixed P_{FA}. Thus, CFAR detectors are, in that sense, reliable. However, deep learning detectors do not classically take into account the constraint on P_{FA}. Neural detectors trained using Dice Loss or Tversky Loss can be calibrated with respect to P_{FA} by applying a threshold filter over the output layer activations : indeed, the network's output layer activations represent the classification of each pixel as a confidence value between 0 (background noise) and 1 (target). Applying a threshold transforms these values into a binary prediction map. In order to determine this threshold, the neural network is then evaluated on a test set S_1 containing varying thermal noise profiles but no target. The pre-threshold confidence indexes of the network are stored and P_{FA} is computed applying a sliding threshold to these values. The threshold is finally selected according to the desired P_{FA}. A representation of this false alarm threshold tuning for the network trained using the baseline Dice loss function is presented in Figure 3.

The research work of [19] shows that it is possible to develop a loss function based on the Neyman-Pearson criterion in order to train a neural network with respect to a fixed constraint, in our case the false alarm probability. The goal of this constrained optimization is to penalize the neural network for producing an unwanted number of false alarms. As such, the constrained detection problem (P) may be formulated as the following :

$$\begin{aligned} \text{minimize}_w\ \mathcal{L}(\hat{y}, y) &= TL(\hat{y}, y) \\ \text{s.t.}\ \ FP(\hat{y}, y) &\leq s \end{aligned} \tag{P}$$

where w are the neural network weights, $TL(\hat{y}, y)$ is the Tversky Loss TL for prediction $\hat{y}$ and ground truth y, and $FP(\hat{y}, y) = \sum_i^N \hat{y}_i(1 - y_i)$. The inequality constraint can be reformulated as an equality one:

$$\text{minimize}_w\ \mathcal{L}(\hat{y}, y) \ = \ TL(\hat{y}, y)$$
$$\text{s.t. } g(\hat{y}, y) = (\frac{1}{s}(FP(\hat{y}, y) - s))^2 = 0$$

This cost function differs from the one used in [19] in the fact that the network is also penalized for producing a number of false alarms that is largely inferior to the threshold. Indeed, setting the penalty to 0 for $FP(\hat{y}, y) < s$ may result in convergence to local minima where both P_{FA} and P_D are close to 0. Particularly, when s is very low, the network may struggle to address the constraint and fall into those local minima. Furthermore, incentivizing the model to produce a number of false alarms closer to the threshold incorporates the idea of threshold calibration that is performed on classical CFAR methods.

The constraint $g(\hat{y}, y)$ may now be incorporated alongside the objective function, in order to go from a constrained problem to an unconstrained one. As such, the updated objective function $\mathcal{L}_P$, using Tversky Loss as a proxy for detection probability, is :

$$\mathcal{L}_P(\hat{y}, y, \lambda) = TL + \lambda g(\hat{y}, y) \tag{3}$$

where λ is an estimate of the Lagrange multiplicator. The neural network trained using Equation 3 will be further referred to as **NN-Lagrange**.

IV. RESULTS

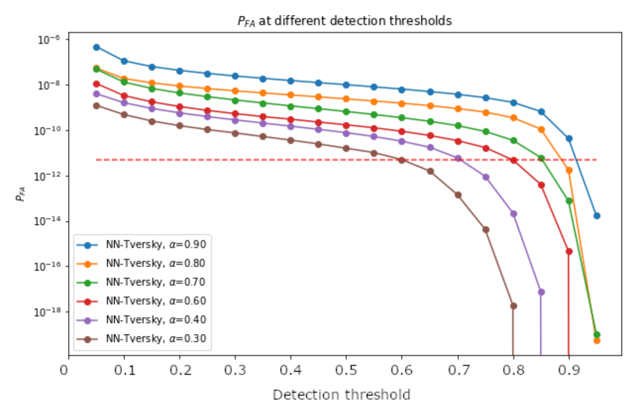

Fig. 4. P_{FA} at various thresholds for NN-Tversky

Figure 4 shows the false alarm probability P_{FA} for different threshold values, for CNN detectors trained using the Tversky loss function with $\alpha \in \{0.1, 0.2, ..., 0.9\}$. It may be observed that, for high values of α, the P_{FA} stays high and only decreases at a large threshold value. The differences between P_{FA} curves for various α is of interest in the setting of radar target detection, as it shall be expanded upon in Section V. For the scenario of this research work, with a singular target on an exoclutter environment, it is decided to choose NN-Tversky with the following hyperparameters : $\alpha = 0.6$ and threshold $= 0.8$.

Figure 5 displays P_{FA} curves against threshold value for the detectors that have been chosen for comparison : NN-Dice, NN-Tversky and NN-Lagrange. P_{FA} for thresholds 0 and 1 are not computed, since values are identical for all models and would hamper figure readability. One may notice that NN-Lagrange exhibits a larger P_{FA} variation across the threshold values in the computed range ($[0.05; 0.95]$) than NN-Dice and

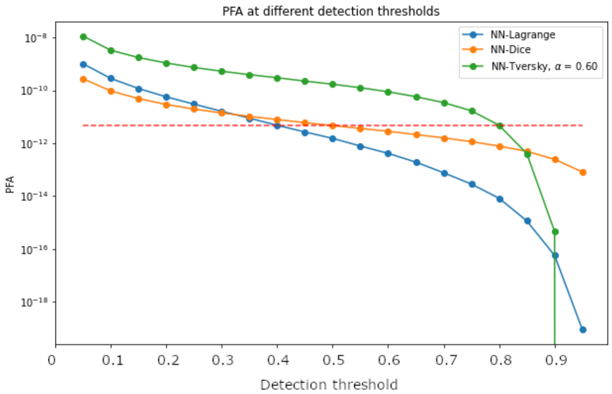

Fig. 5. P_{FA} at various thresholds

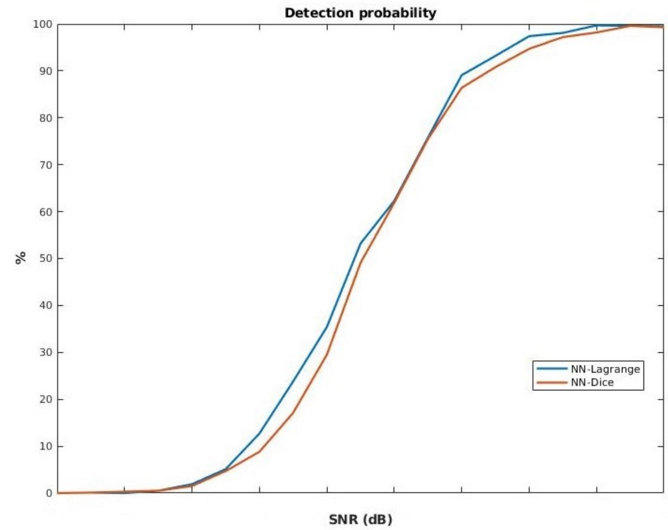

Fig. 7. Comparison of the inclusion of TL vs DL in the constrained objective function

NN-Tversky - that is, NN-Lagrange can produce a higher range of P_{FA} values by threshold setting than the other loss functions which seem limited to a narrower range. The final layer activation threshold values are chosen according to the expected number of false alarms.

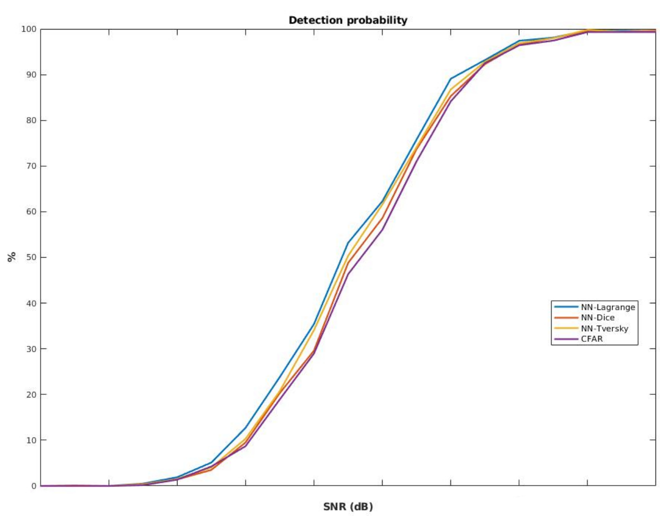

Fig. 6. Detection probability as a function of SNR

Model	CFAR	NN-Dice	NN-Tversky	NN-Lagrange
$P_{FA}(\times 10^{-7})$	1.64	0.951	1.15	0.573

TABLE I
P_{FA} VALUES FOR DIFFERENT DETECTORS

It can be seen, from Figure 6, that NN-Lagrange provides the best detection performances across all methods. All neural detectors improve P_D over the CFAR detector, which detection probability is decreased by the CFAR loss resulting from a potentially imprecise noise variance estimation due to the limited number of noise samples. NN-Lagrange noticeably outperforms NN-Tversky and NN-Dice. Indeed, while the latter two have seen their final layer activation threshold fine-tuned using detection on targetless thermal noise, NN-Lagrange incorporates the constraint on the number of false alarms during training. It is then straightforward to note that there is less information loss with NN-Lagrange. It may also be seen that NN-Tversky boasts a slightly higher detection performance than NN-Dice. Again, this is due to the fact that the network has been trained to produce a higher number of detections, which can be leveraged by fine-tuning the detection threshold.

Table I shows the false alarm probability P_{FA} for the compared models. It can be appreciated that all neural network detectors exhibit a lower P_{FA} than CFAR. P_{FA} values gravitate around the same operating point, with the exception of NN-Lagrange, which remarkably achieves both the highest P_D and the lowest P_{FA}. The relative improvements of the studied loss functions over CFAR can thus be validated.

Ablation study

An ablation study is performed in order to understand the choice of the Tversky Loss TL instead of the Dice Loss DL in the objective function for the constrained optimization problem (P). Two networks are compared : NN-Lagrange, which has been introduced earlier, and a neural network trained by replacing the Tversky Loss TL by the Dice Loss DL :

$$\mathcal{L}_P(\hat{y}, y, \lambda) = DL + \lambda g(\hat{y}, y) \quad (4)$$

Figure 7 shows that, for an identical P_{FA}, the network trained using TL in the objective function displays a higher P_D than the detector trained with DL. This difference in P_D is explained by the fact that the former network proposes more detection candidates due to the nature of TL when α is close to 1. This higher number of detection candidates has no influence on P_{FA} due to the constraint.

V. Discussion

As it has been discussed in Section IV, NN-Tversky shows, at a fixed P_{FA}, higher detection performance than NN-Dice. This superiority may be explained by the fact that the parameter α Tversky Loss incorporates provides an additional degree of liberty during the network calibration process. One is able to select both the optimal α and the optimal final layer activation threshold in order to maximize detection probability at a fixed P_{FA}. This versatility can also be used to calibrate the network on other more difficult environments, containing perturbations and/or interactions. However, selecting an optimal value of α requires network retraining, while selecting an optimal threshold is performed after training. Whether model training for a specific α may be performed in little time, using, e.g., transfer learning, will be the subject of further investigations.

However, the hyperparameters yielding optimal performance for NN-Tversky are empirically found and tuned and are not backed by theoretical guarantees. NN-Lagrange addresses this shortcoming by bringing a notion of stability, as the network is trained to respect a given P_{FA}. However, we show that the constraint, while satisfied during training, is not guaranteed to be at inference, because the loss function is not computed. Giving theoretical P_{FA} guarantees for CNN detectors at inference time is still the object of research.

It is interesting to notice that, even though the network has been trained on a dataset containing thermal noise as well as ground clutter, in hopes of having a model capable of generalizing to complex environments, the detection performances do not decrease compared to a network that has been trained on thermal noise alone. This can be explained by the fact that adding ground clutter or other perturbations brings robustness to the model.

VI. Conclusion

In this paper, a deep learning model trained to perform radar target detection is proposed. Original loss functions that are not commonly used in deep learning-based model training for radar signal processing are introduced and their contributions to improvements on detection probability and false alarm control are highlighted.

The detection performances of the studied networks are determined on a singular target detection over exoclutter background scenario, which is the environment on which the performance gains are supposed to be minimal compared to the classical CFAR methods. It is shown that the neural detectors outperform the CFAR baseline in this setting. The next step, detection performance evaluation over more complex environments (presence of ground clutter, interferences, multiple targets, real data) will be the object of future research. Improvements of using neural detectors over CFAR detectors are expected to be much greater, as it has been noticed through early experiments.

References

[1] Louis L. Scharf and Cédric Demeure, *Statistical signal processing: detection, estimation, and time series analysis*, Addison-Wesley series in electrical and computer engineering. Digital signal processing. Addison-Wesley Pub. Co, Reading, Mass, 1991.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. Burges, L. Bottou, and K. Q. Weinberger, Eds. 2012, vol. 25, Curran Associates, Inc.

[3] Karen Simonyan and Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 2014.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep Residual Learning for Image Recognition," 2015.

[5] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully Convolutional Networks for Semantic Segmentation," 2014.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," 2015.

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, "Mask R-CNN," 2017.

[8] P.P. Gandhi and S.A. Kassam, "Analysis of CFAR processors in nonhomogeneous background," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 24, no. 4, pp. 427–445, July 1988.

[9] Chao Wang, Jian Wang, and Xudong Zhang, "Automatic radar waveform recognition based on time-frequency analysis and convolutional neural network," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, Mar. 2017, pp. 2437–2441, IEEE.

[10] Bharat Sehgal, Hanumant Singh Shekhawat, and Sumit Kumar Jana, "Automatic Target Recognition Using Recurrent Neural Networks," in *2019 International Conference on Range Technology (ICORT)*, Balasore, India, Feb. 2019, pp. 1–5, IEEE.

[11] Daniel Brodeski, Igal Bilik, and Raja Giryes, "Deep Radar Detector," in *2019 IEEE Radar Conference (RadarConf)*, Boston, MA, USA, Apr. 2019, pp. 1–6, IEEE.

[12] Li Wang, Jun Tang, and Qingmin Liao, "A Study on Radar Target Detection Based on Deep Neural Networks," *IEEE Sensors Letters*, vol. 3, no. 3, pp. 1–4, Mar. 2019.

[13] Faruk Yavuz, "Radar Target Detection with CNN," in *2021 29th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 1581–1585.

[14] Chenxing Wang, Jiangmin Tian, Jiuwen Cao, and Xiaohong Wang, "Deep Learning-Based UAV Detection in Pulse-Doppler Radar," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–12, 2022.

[15] Meiyan Pan, Jianjun Chen, Shengli Wang, and Ziwei Dong, "A Novel Approach for Marine Small Target Detection Based on Deep Learning," in *2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP)*, Wuxi, China, July 2019, pp. 395–399, IEEE.

[16] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," 2015.

[17] Tzvi Diskin, Uri Okun, and Ami Wiesel, "Learning to Detect with Constant False Alarm Rate," 2022.

[18] Chia-Hung Lin, Yu-Chien Lin, Yue Bai, Wei-Ho Chung, Ta-Sung Lee, and Heikki Huttunen, "DL-CFAR: A Novel CFAR Target Detection Method Based on Deep Learning," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, 2019, pp. 1–6.

[19] Zachary Baird, Michael K. McDonald, Sreeraman Rajan, and Simon Lee, "A Neyman-Pearson Criterion-Based Neural Network Detector for Maritime Radar," in *2021 IEEE 24th International Conference on Information Fusion (FUSION)*, Sun City, South Africa, Nov. 2021, pp. 1–8, IEEE.

[20] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," 2016.

[21] Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour, "Tversky loss function for image segmentation using 3D fully convolutional deep networks," 2017.

[22] Roberto Franceschi and Dmytro Rachkov, "Deep learning-based radar detector for complex automotive scenarios," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022, pp. 303–308.