Radar Target Detection with CNN

Faruk Yavuz
ASELSAN Research Center
ASELSAN Inc.
Ankara, Turkey
farukyavuz@aselsan.com.tr

***Abstract*—Target detection is a fundamental radar application that is traditionally carried out by Constant False Alarm Rate (CFAR) detectors. This paper proposes a Convolutional Neural Network (CNN) based detector (RadCNN) to replace the standard CFAR detectors for a typical pulsed Doppler radar. RadCNN takes patches of the range-Doppler ambiguity function as input and returns detection status for the input patch. A radar simulator is developed for data generation with desired noise and clutter scenarios. RadCNN is compared against Cell-Averaging (CA), Smallest of Cell Averaging (SOCA), Greatest of Cell Averaging (GOCA), Ordered Statistics (OS) CFAR and similar state of the art detectors in the literature. The comparison is done for a variety of scenarios including multiple targets, thermal noise and clutter at different Signal to Noise Ratios (SNR) and Clutter to Noise Ratios (CNR). It is shown that RadCNN improves the performance of CFAR for low SNR and exhibits four orders of magnitude less computational complexity than the similar state of the art and realizable in real-time applications.**

Keywords—Deep Learning, Radar, Target Detection, CNN, CFAR.

I. Introduction

Radar has been widely used for military and civilian applications since 1904 [1]. Radars address a diverse set of applications including surveillance, navigation, monitoring, mapping, weather forecasting, and collision avoidance [1], [2]. Pulse-Doppler radar, in particular, scans the surrounding environment by transmitting pulses from an antenna or an array of antennas, and processes the received echoes of these pulses to perform signal conditioning, detection, and post-detection tasks like tracking or target recognition [2].

Pulse-Doppler radar arranges the received echoes into range-Doppler data matrix, which is further processed to obtain the ambiguity function over a grid of range and Doppler axes [2]. The system checks for targets by comparing the reflected signal strength against a threshold derived from the data matrix. Unpredictable nature of interference requires the use of adaptive algorithms like Constant False Alarm Rate (CFAR) detectors [2]. CFAR detectors differ in calculation of the threshold, the most popular ones being Cell-Averaging (CA), Smallest of CA (SOCA), Greatest of CA (GOCA), and Ordered Statistics (OS) CFAR algorithms [2].

Deep Learning has become quite popular over the past decade for automating the feature learning process. The features are learnt from large datasets by scanning different abstraction layers. Speech recognition, object detection, and natural language processing are some of the fields where Deep Learning has advanced the state of the art remarkably [3]. One of the many Deep Learning techniques is Convolutional Neural Networks (CNNs). CNNs use convolutional filters to extract the underlying features of the data and perceptron layers for the classification using the extracted features [4]. CNNs have achieved significant performance improvements in image classification tasks starting with 'LeNet-5' [4]-[7].

Mason *et al.* have forecasted the application of Deep Learning to radar [8]. Chen *et al.* showed that CNN based algorithms outperform the existing Synthetic Aperture Radar (SAR) target recognition algorithms [9]. Akçakaya *et al.* applied incremental learning to radar [10]. Deng *et al.* performed target detection using decision trees with feature vectors and machine learning [11]. Wang *et al.* applied a 'LeNet-5' inspired CNN detector for radar [12]. Following the work in [12], Xie *et al.* showed that the CNN improved the performance on real data [13]. Yavuz *et al.* developed a CNN detector for single target in homogeneous interference [14].

This paper proposes a CNN based radar multi-target detection algorithm (RadCNN). RadCNN takes patches of the range-Doppler ambiguity function as its input and generates probability of target presence at the center of the patch. RadCNN employs a different classification method and a different network structure compared to the detectors given in [12], [13]. RadCNN learns to detect targets from the training patches of the range-Doppler input and slides through the input for detection at every possible cell, like [12]. However, RadCNN uses one convolutional layer and fully connected layer whereas [12] employs multiple convolutional layers and two fully connected layers. RadCNN works in the presence of noise and clutter, whereas [12] only works under noise.

A radar simulator to emulate radar scenarios with desired levels of clutter and thermal noise is developed for data. Detector performances are evaluated in terms of probability of detection (P_D), probability of false alarm (P_{FA}) and computational complexity. Since [13] implemented [12] for real radar data, the performance of [12] is studied in this study. The performance results of [12] are retained here in order to make the comparisons bias-free.

The rest of the paper is organized as follows. Section II reviews the traditional CFAR algorithms. Section III details the proposed technique and adopted target scenarios. Section IV presents the simulation results. Section V concludes the proposed work and lays out future work.

II. Constant False Alarm Rate Detection

CFAR algorithms compare the amplitude of the received signal to an adaptive threshold. The threshold is set to maintain a constant false alarm rate assuming that the interference follows a certain distribution. Adhering to CFAR terminology, the unit to be tested is the Cell-Under-Test (CUT), the cells used for estimation of the interference are Reference Cells (RCs), and the cells between the CUT and the RCs are the Guard Cells (GCs). CFAR window shifts through the range-Doppler data and tests each cell for a target. When both range and Doppler cells are used as RCs, CFAR is called two-dimensional (2D). The flow of a typical 2D CFAR algorithm is shown in Fig. 1.

The 2232 International Fellowship for Outstanding Researchers program of TÜBİTAK supports this work (Project No: 118C253). The author holds all the responsibility for the content of the paper. Financial support does not mean that the content is approved in a scientific sense by TÜBİTAK.

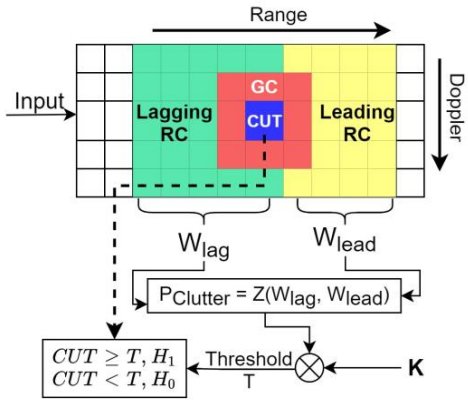


Fig. 1. 2D CFAR algorithm.

As shown in Fig. 1, clutter strength ($P_{Clutter}$) is estimated using the RCs, and the threshold is calculated by multiplying the former with a scaling constant K. A target is detected if signal strength of CUT is greater than $KP_{Clutter}$. CFAR algorithms differ in the estimation of $P_{Clutter}$ and K. Each CFAR detector has pros and cons depending on the radar environment. Common CFAR algorithms are detailed in the following subsections. The computational complexities are summarized for a data matrix having N_R and N_D cells along the range and Doppler axes, respectively.

A. CA-CFAR

CA-CFAR is developed for operation in homogeneous interference scenarios [2]. $P_{Clutter}$ is estimated as the mean of all available RCs. K is calculated using the total number of RCs (N) for a square-law detector as

$$K = N\,({P_{FA}}^{-1/N} - 1)\ . \tag{1}$$

CA-CFAR performs poorly at clutter transitions, under target masking scenarios and for nonhomogeneous interference [2]. Computational complexity is given as $O(N_R N_D N)$.

B. SOCA-CFAR

SOCA-CFAR is developed to handle target-masking scenarios [2]. $P_{Clutter}$ is estimated as the smaller of the means of lagging and leading RCs. For a square-law detector, K is calculated as

$$P_{FA} = 2\sum_{k=0}^{\frac{N}{2}-1}\binom{\frac{N}{2}-1+k}{k}\left(2+\frac{K}{N/2}\right)^{-\frac{N}{2}-k}. \tag{2}$$

SOCA-CFAR outperforms CA-CFAR under target masking scenarios, but yields a high CFAR loss for other scenarios [2]. Computational complexity is given as $O(N_R N_D (N+1))$.

C. GOCA-CFAR

GOCA-CFAR algorithm is developed to handle clutter edge transitions [2]. $P_{Clutter}$ is estimated as the greater of the means of lagging and leading RCs. For a square-law detector, K is calculated as

$$P_{FA} = 2\left(1+\frac{K}{N/2}\right)^{-\frac{N}{2}} - 2\sum_{k=0}^{\frac{N}{2}-1}\binom{\frac{N}{2}-1+k}{k}\left(2+\frac{K}{N/2}\right)^{-\frac{N}{2}-k} \tag{3}$$

GOCA-CFAR outperforms CA-CFAR at clutter edges, but fails under target masking scenarios and exhibits additional CFAR loss for homogenous interference [2]. Its computational complexity is given as $O(N_R N_D (N+1))$.

D. OS-CFAR

OS-CFAR algorithm is developed to improve CA-CFAR at clutter transitions and under target masking scenarios [2]. $P_{Clutter}$ is estimated as a certain rank of RCs. The rank representing $P_{Clutter}$ is chosen as 75% of N [2]. K is calculated for a square-law detector as

$$P_{FA} = k\binom{N}{k}B(K+N-k+1,\ k) \tag{4}$$

where k is the rank and $B(.)$ is the beta function. OS-CFAR combines the strengths of CFAR detectors discussed above and improves the overall detection performance with a reasonable CFAR loss at the cost of extra computations [2]. The complexity becomes $O(N_R N_D N \log N)$.

III. Target Detection with RadCNN

CNN algorithms have good performance on multi-label image classification problems [5]-[7]. Inspired by this principle, a CNN is designed to take patches of the range-Doppler ambiguity matrix as its input and to classify target presence. The dataset and RadCNN are detailed in the following subsections.

A. Data Preperation

A radar simulator is developed in order to generate the data required to test RADCNN and traditional CFAR algorithms. The simulated radar operates at 1 GHz carrier modulated by linear frequency modulated (LFM) pulses with 7.5 MHz bandwidth and 20 µs pulse duration over a coherent processing interval (CPI) of 16 pulses. Range and velocity resolutions translate to 20 m and 9.38 m/s, respectively, with an accompanying range bin spacing of 10 m. Over the unambiguous range swath and speed interval of 30 km and 75 m/s, the generated range-Doppler data matrix has a size of 3000 x 16. Simulated scenario features Swerling type-0/5 targets, independent identically distributed (i.i.d.) white noise, and clutter. Following clutter settings are employed [2]:

- Weibull distributed clutter over 5-7.5 km and around zero Doppler with shape and scale parameters of 1 and 1.
- Weibull distributed clutter over 7.5-10 km and around zero Doppler with shape and scale parameters of 2 and 1.
- K-distributed clutter over 15-17.5 km and around zero Doppler with shape and scale parameters of 3 and 1.
- K-distributed clutter over 17.5-20 km and around zero Doppler with shape and scale parameters of 15 and 1.

The clutter settings are designed to realize clutter walls, noisy area to clutter transition, and clutter to clutter transitions within the CFAR reference window. Clutter to clutter transitions are simulated by varying parameters within the same distribution. Weibull and K-distributed clutters are commonly used for simulating land and sea clutter, respectively [2]. Radar detection is casted as a classification task by assigning two labels to data patches, indicating the target presence at the center of the patch. These patches are chosen to represent challenging and common target detection scenarios. Specifically, the scenarios demonstrate cases with targets under clutter with CFAR window overlapping the noise region (S1), targets at the center of clutter (S2), targets in noise region with CFAR window overlapping the clutter (S3), and targets in noise region only (S4).

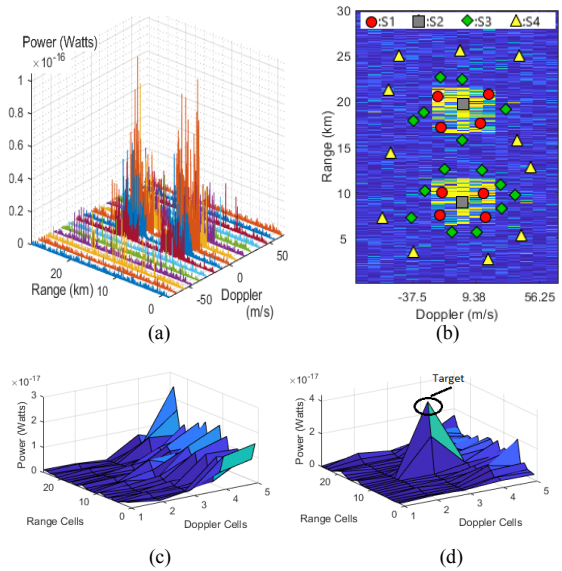


Fig. 2. (a) Sample range-Doppler image, (b) Target scenarios, (c) Target absent patch, (d) Target present patch

CNN input is designed as the patches of size 25 x 5 from the squared magnitude ambiguity matrix. The ambiguity matrix is constructed by matched-filtering the received signal in range and by subsequently taking FFT of the matched-filtered signal in Doppler space [2]. Fig. 2(a) presents a sample range-Doppler image and Fig. 2(b) highlights example positions for test scenarios. Fig. 2(c) shows target absent patch example and Fig. 2(d) shows target present patch example.

The training data comprises patches of range-Doppler images simulating Clutter to Noise Ratios (CNRs) of 4 dB, 7 dB, and 10 dB. For each CNR, 200 range-Doppler images are created and are cut into 45500 patches per the aforementioned detection scenarios as target absent dataset. Then, each patch is simulated with a target at the center of the patch for target present data set at SNR values of 13 dB, 16 dB and 20 dB. The entire training data includes the target absent set and target present set at three different SNR, a total of 182000 files. Generated data are split into training, validation and test sets with ratios of 70-15-15%, respectively.

B. *Proposed Detector*

In order to improve the CNN detector given in [12], generic CNN structure is changed. Considering the domain

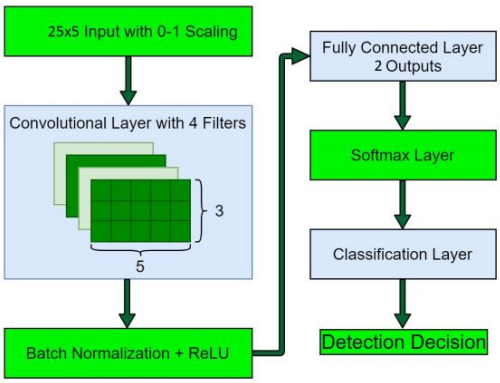


Fig. 3. Proposed CNN model.

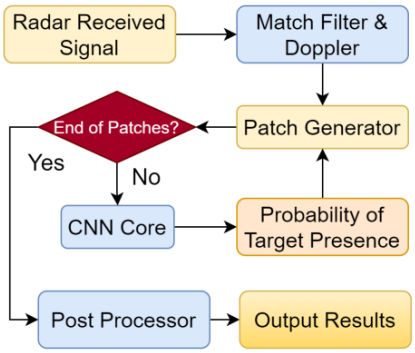


Fig. 4. RadCNN algorithm flow

characteristics, CNN is designed as a shallow network. Fig. 3 illustrates the developed CNN.

Stochastic gradient descent with momentum is used for the training of the network at a learning rate of 0.01. Batch size and momentum is set to 128 and 0.9, respectively. As shown in Fig. 3, the network consists of an input layer with zero-one scaling, a convolution layer with 4 filters of size 5x3, a batch normalization layer, a ReLU layer, a fully connected layer with 2 outputs, a softmax layer, and a classification layer. The CNN achieved 87.41%, 87.42%, and 87.4% accuracy on the training, validation, and test data, respectively. The trained CNN is used as the core of RadCNN as shown in Fig. 4.

As observed from Fig. 4, the received radar signal is matched-filtered in range and an FFT is subsequently applied along the Doppler axis. The resulting range-Doppler ambiguity matrix is magnitude squared for square-law detection. Patch generator slides through the range-Doppler image and generates 25x5 sized patches with a stride of 1. The CNN runs on the patch and decides the probability of target presence for the patch. This probability is registered onto the output matrix of size 3000x16. The algorithm enters the post processing stage once patch generator visits all the cells in the range-Doppler image. Post processor compares the probability matrix with a threshold. If the probability exceeds the threshold, RadCNN detects a target at the corresponding cell. The selection of the threshold determines the P_{FA} of RadCNN. Hence, RadCNN has a tunable P_{FA} and employs only a single convolutional layer, contrasting with [12].

IV. Simulation Results

Performance of RadCNN is analyzed with simulated data. The test data is simulated using the training scenario using SNR values of 0, 1, 3, 6, 9, 11, 13, 15, 17, 20 dB each with CNRs of 4 and 10 dB. Simulations were run without a target and with ninety-eight targets. The targets are distributed in the range-Doppler image per the aforementioned four scenarios for each simulation. Hence, the experiments cover 40 test configurations with multiple examples of all scenarios tested with 200 Monte Carlo iterations.

CFAR algorithms have a constant P_{FA} of 10^{-3} with 20x2 RCs and 4x2 GCs along range-Doppler axes. OS-CFAR has a rank ratio of 0.75. RadCNN post processor threshold is set as 0.65 to acquire the P_{FA} of 10^{-3}. The results are analyzed in five different categories. The categories are designed to compare RadCNN to the CFAR detectors and [12] in realistic and demanding radar environments.

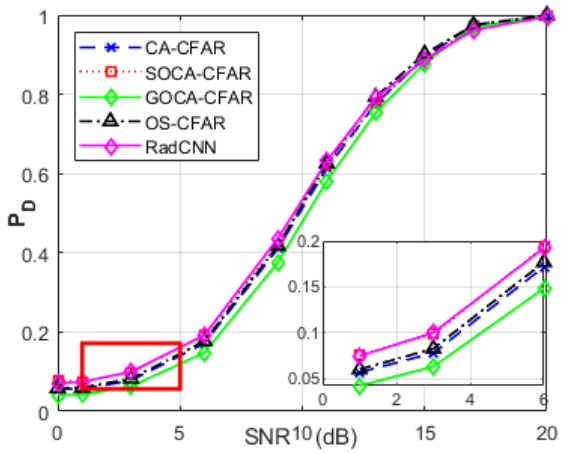


Fig. 5. Average P_D of all simulations.

Fig. 5 shows the average P_D calculated using all simulations and all scenarios. Table I lists the average P_{FA} calculated from Monte Carlo runs and the average computational cost for each algorithm. Fig. 6 and Fig. 7 show the simulated P_D as a function of SNR for each of the four scenarios S1-S4 at a given CNR.

As one observes from Fig. 5, RadCNN improves the performance of CFAR detectors by approximately 0.5 dB at low SNR values. RadCNN performance is comparable to CFAR detectors at high SNR values. It should be noted that this performance is a result of all the simulated scenarios and environments. It is also important to analyze the performance for specific scenarios and environments.

Table I shows that average P_{FA} of RadCNN is $1.5\text{x}10^{-3}$ and the CFAR algorithms cannot maintain their preset P_{FA} of 10^{-3}. In agreement with the literature, SOCA-CFAR has the highest P_{FA} followed by OS-CFAR, CA-CFAR and GOCA-CFAR [2] due to the presence of clutter. In addition to the fact that RadCNN has much smaller computational cost compared to [12] (by four orders of magnitude), its performance for non-homogeneous and homogeneous mixture environment is better than the CFAR detectors and [12].

Authors of [12] supplied the results for noise only scenario and experimented with data patches where target is at the center. According to the reported performance of [12], the model outperforms CA-CFAR by 0.2 dB for high SNR and the model has comparable performance for low SNR. Authors of [12] did not report numerical values for CA-CFAR P_{FA}. On the contrary, RadCNN is tested against different types of clutter with various configurations and the test patches are

TABLE I. DETECTOR PERFORMANCE COMPARISON.

Detector Type	*Average P_{FA}*	*Average Operation Count* [a]
CA-CFAR	$1.3\text{x}10^{-3}$	$57.6\text{x}10^{5}$
SOCA-CFAR	$1.7\text{x}10^{-3}$	$58.1\text{x}10^{5}$
GOCA-CFAR	$1.2\text{x}10^{-3}$	$58.1\text{x}10^{5}$
OS-CFAR	$1.6\text{x}10^{-3}$	$39.8\text{x}10^{6}$
RadCNN	$1.5\text{x}10^{-3}$	$49.1\text{x}10^{7}$
[12]	Not Applicable	$54.2\text{x}10^{11}$

[a] Multiply, add, compare operations.

shifted across the range-Doppler image by one cell. Hence, RadCNN also needs to differentiate cases where the target is at the center of the patch by declaring target present and cases where target is off the center by declaring target absent. Despite the more comprehensive and challenging tests, RadCNN provided more improvement than [12] and has 10000x less computational complexity.

Computational savings of RadCNN over [12] originates from the reduced CNN depth, smaller input size, dismissal of second fully connected layer as well as the dramatic decrease on the number of convolution kernels.

One notes from Table I that the computational complexity of RadCNN is still approximately 10x the complexity of OS-CFAR and 80x the complexity of CA-CFAR. Parallel GPU implementation of similar networks usually result in more than 10x speed improvements compared to the CPU implementations [15]. Hence, RadCNN is realizable in a real world scenario whereas [12] is far from a real-time realization.

Since traditional CFAR algorithms are designed to work better under specific conditions, as the next step, RadCNN is compared against CFAR algorithms under these particular scenarios. Note that since [12] is not tested in the presence of clutter, a comparison is not possible.

As seen on Fig. 6(a), RadCNN performs better than the CFAR algorithms, especially at low SNR when target is at the clutter border but within the clutter. Even though the performance gap between RadCNN and the rest decrease as CNR increases, RadCNN stays robust at high CNR scenarios.

One observes from Fig. 6(b) that RadCNN outperforms the CFAR algorithms when target is covered by clutter. As CNR increases, the performance improvement of RadCNN becomes more distinct by exceeding a gain of 1 dB.

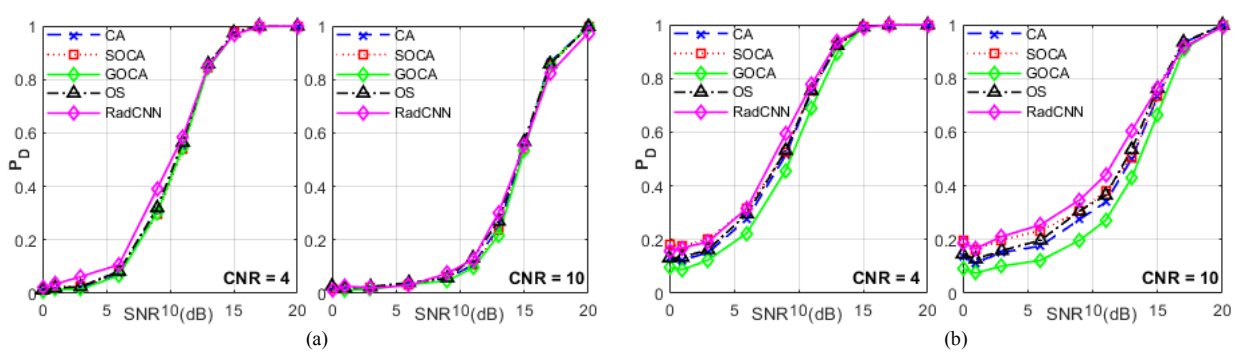
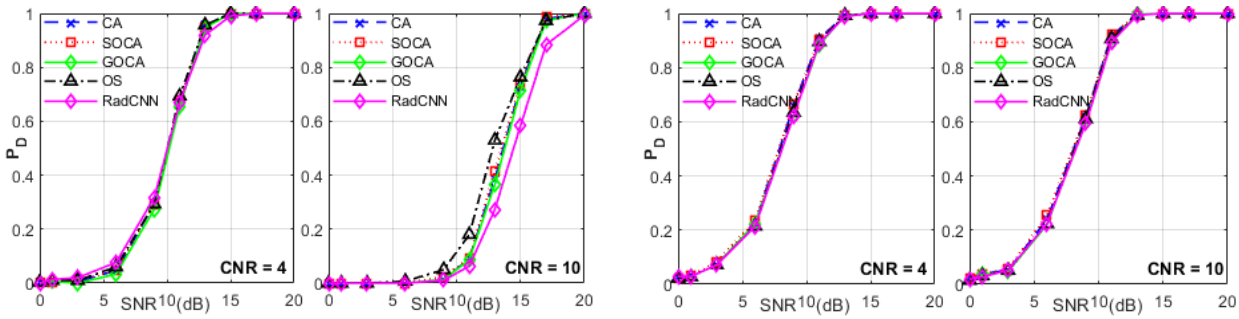


Fig. 6. Simulated P_D vs SNR: (a) Scenario S1, (b) Scenario S2



(a) (b)

Fig. 7. Simulated P_D vs SNR: (a) Scenario S3, (b) Scenario S4.

Fig. 7(a) shows that RadCNN is not as successful at detecting targets when they are near clutter but the reference window is mostly occupied by noise. At low CNR values, RadCNN performance is slightly better than the CFAR detectors but as CNR increases, RadCNN starts to perform worse than the traditional algorithms by almost 0.8 dB. This scenario will be the main challenge for future work.

Fig. 7(b) shows that RadCNN performs on par with the CFAR algorithms when the target is surrounded by only noise. It should be noted that the CFAR techniques perform quite similarly but RadCNN has approximately 10% less P_{FA} than OS and SOCA CFAR algorithms.

V. Conclusion

Target detection is the most fundamental application of the modern radar. Various types of CFAR detectors carry out target detection specializing at specific scenarios. This paper proposes RadCNN, a CNN based multi-target detector, to replace the traditional CFAR algorithms. Simulations under different scenarios including clutter transitions and clutter walls demonstrate that RadCNN improves the performance of the traditional CFAR algorithms and outperforms the similar state of the art CNN based radar target detector [12]. Even though RadCNN is tested more rigorously with different types of clutter distributions and target detection scenarios than [12], RadCNN achieves better performance improvement over the CFAR techniques. It is important to emphasize that RadCNN has four orders of magnitude less computational complexity than [12], meaning that it is realizable for real-time radar applications. In the light of the presented simulation results, it becomes clear that RadCNN detector is a viable and superior alternative to the similar state of the art [12] for realistic radar scenarios comprising both homogeneous and nonhomogeneous interference.

It is planned to extend the presented work with a number of follow-up studies such as tests with measured radar data, improvements for additional computational savings, and possible improvements for clutter wall scenarios. Especially, the future studies will focus on the clutter wall scenarios where the target is in the noisy region. If RadCNN is further improved for target near clutter scenarios while maintaining performance for the rest, it will be an overall superior detector that works for all challenging radar scenarios.

Acknowledgment

The author thanks Dr. Çağrı Çetintepe for his valuable contributions during the review process.

References

[1] O. Blumtritt, H. Petzold and W. Aspray, Tracking the History of RADAR, IEEE Rutgers, New Jersey, 1994.

[2] M. A. Richards, Fundamentals of Radar Signal Processing, *1st ed.*, New York, NY, USA: McGraw-Hill, 2005.

[3] Y. LeCun, Y. Bengio and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.

[4] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, *1st ed.*, Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org.

[5] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient Based Learning Applied to Document Recognition," *Proceedings of IEEE*, vol. 86, no. 11, pp.2278-2324, 1998.

[6] N. Aloysius and M. Geetha, "A review on deep convolutional neural networks," *2017 International Conference on Communication and Signal Processing (ICCSP)*, Chennai, 2017, pp. 0588-0592.

[7] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1106–1114.

[8] E. Mason, B. Yonel and B. Yazici, "Deep learning for radar," *2017 IEEE Radar Conference (RadarConf)*, Seattle, WA, 2017, pp. 1703-1708.

[9] S. Chen, H. Wang, F. Xu and Y. Q. Jin, "Target Classification Using the Deep Convolutional Networks for SAR Images," *IEEE Trans. Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4806-4817, 2016.

[10] M. Akcakaya, S. Sen and A. Nehorai, "A Novel Data-Driven Learning Method for Radar Target Detection in Nonstationary Environments," *IEEE Signal Processing Letters*, vol. 23, no 5, pp. 762-766, 2016.

[11] H. Deng, Z. Geng and B. Himed, "Radar Target Detection Using Target Features and Artificial Intelligence," *2018 Int. Conf. on Radar (RADAR)*, Brisbane, QLD, 2018, pp. 1-4.

[12] L. Wang, J. Tang and Q. Liao, "A Study on Radar Target Detection Based on Deep Neural Networks," in *IEEE Sensors Letters*, vol. 3, no. 3, pp. 1-4, March 2019.

[13] Y. Xie, J. Tang and L. Wang, "Radar Target Detection using Convolutional Neutral Network in Clutter," *2019 IEEE Int. Conf. on Signal, Information and Data Processing (ICSIDP)*, Chongqing, China, 2019, pp. 1-6.

[14] F. Yavuz and M. Kalfa, "Radar Target Detection via Deep Learning," *2020 28th IEEE Conf. on Signal Processing and Communications Applications (SIU)*, 2020, pp. 1-4.

[15] S. Singh, A. Paul and M. Arun, "Parallelization of digit recognition system using Deep Convolutional Neural Network on CUDA," *2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS),* Chennai, 2017, pp. 379-383.