Deep Convolutional Autoencoder Applied for Noise Reduction in Range-Doppler Maps of FMCW Radars

Marcio L. Lima de Oliveira
Computer Architecture for Embedded Systems
University of Twente
Enschede, The Netherlands
Email: m.l.limadeoliveira@utwente.nl

Marco J. G. Bekooij
Department of Embedded Software and Signal Processing
NXP Semiconductors
Eindhoven, The Netherlands
Email: marco.bekooij@nxp.com

***Abstract*—In this paper, we discuss the usage of deep Convolutional Autoencoders (CAE) for denoising Range-Doppler (RD) maps of an FMCW radar in a near-field situation with pedestrians and cyclists as moving objects. Traditional methods for noise reduction such as CFAR with Peak Detection (PD) have poor performance under highly noisy environments, especially when objects have low reflectivity, like pedestrians. We propose the use of Convolutional Autoencoders, in various configurations, to overcome those limitations. Due to its Artificial Neural Network nature, CAE can extract features, learn to identify patterns and recognize moving objects - such as humans and bicycles - while the traditional method acts passively. The results indicate that the usage of CAE overcame CFAR with PD by 76.8%, on average, for reconstructing objects in their correct positions. The results of this work may have many applications, for instance, detecting distant moving objects that are too faint for regular detection.**

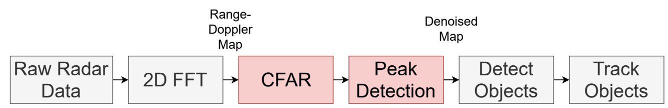
I. Introduction

The future road surveillance, as well for autonomous vehicles, is going towards multi-sensor systems capable of detecting, classifying and tracking objects such as pedestrians, cyclists and cars. However, cameras depend highly on good lighting, and laser sensors (such as LiDAR) are expensive and fragile. Radars, on the other hand, are cheap, robust and can work regardless of weather or light conditions.

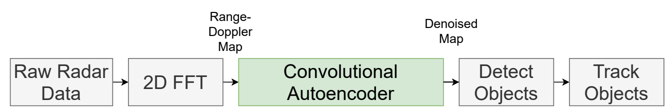
Radars have been commonly used for decades and nowadays applications can vary fairly: from the classification of a cooking process [1] to military application [2]. Nevertheless, the employment of Deep Learning (DL) in this field is just beginning, though it has become a common approach for image processing. Exploring the usage of artificial intelligence on radar applications may be beneficial since it has similarities with image processing.

Constant False Alarm Rate (CFAR) - or its variations - is a commonly employed technique to determine the noise frequency level, and identify possible objects in Range-Doppler maps when followed by a Peak Detector (PD). Some papers using CFAR and PD can be found in [3]–[5]. Besides its adaptive capabilities over fixed thresholds, CFAR performs poorly when the noise level is high, the target object is not

This work is funded by the NWO Program ZERO.



(a) Traditional object tracking for radars scheme



(b) Proposed change

Fig. 1. Simplified schematics of the traditional tracking and the proposed change.

much reflective, or the radar is moving. It also needs constant re-calibration [6].

Frequency-Modulated Continuous-Wave (FMCW) radars can directly measure the distance to an object and their relative radial velocity. With this data, it is possible to build a Range-Doppler (RD) map. This map is a two-dimensional heat map, where one axis represents the distance between radar and object; and the other represents the relative radial speed of them. An RD map is a matrix, in which each element corresponds with the normalized received reflected power (in decibels) of a detected object at a certain speed and range; hence, this map can be treated as an image and, by doing so, it is possible to apply many known image processing techniques to it, such as Autoencoders (AE). More about Range-Doppler maps can be found in [7]. A deep Convolutional Autoeconder (CAE) is a Convolutional Neural Network (CNN) capable of reconstructing images. Applications with it can vary: pose classification [8], noise filtering [9], or even multi-spectral image fusion [10].

In this paper, we propose a novel technique to reduce noise in Range-Doppler maps utilizing CAE instead of a traditional CFAR method. The advantages for doing this are various: CAE outperforms CFAR, especially in highly noisy situations; CAE learns patterns and can discern between noise and a moving object, therefore it is more robust.

In the related work of Seyfioglu et al. [8], they have applied

CAE to classify 12 distinct human activities - including walking, jogging, running, crawling, sitting, among others - using micro-Doppler spectrograms extracted from radars. According to their paper, they could classify correctly the corresponding activity with 94.2% of accuracy. They claim their results are 17.3% superior to using Support Vector Machines.

Fig. 1 shows a commonly used schematics for tracking an object when using FMCW radars. Similarly is what companies - like Texas Instruments [11] - and other researchers [12] employ. Our goal during this work is to modify the portion of the traditional denoising technique by applying a CAE.

II. Deep Convolutional Autoencoder Architectures and Configurations

Autoencoders are, usually, an unsupervised algorithm that take a map from a higher-dimensional space to a lower-dimensional with the end goal that the first map can be recreated (approximately) from the lower-dimensional portrayal [13]. In other words, it tries to recreate the input by lowering its dimensions to extract features.

A Deep Convolutional Autoencoders (CAE) utilizes Convolutional Neural Networks (CNN) in a deep learning scheme to extract features from images or image-like objects. Like any other autoencoder, it has an encoder where the feature extraction occurs, and a decoder where the image is reconstructed. We chose the architecture shown in Table I after many experiments using various parameter settings (Section IV-A). In this table it is possible to see the layers of the CNN used to create both encoder and decoder.

During the training process, CAE uses a noisy image and its counterpart, a denoised image, as the input. For this work, we have used Mean Squared Error (MSE) as the loss function and Root Mean Squared Propagation (RMSprop) as the optimizer that tries to minimize the loss function to correct the weights and biases of the CAE.

Following Fig. 2, it is possible to see what occurs during the evaluation of an already-trained network. The encoder receives a noisy RD map featuring two people walking towards the radar to extract features and important shapes. The extracted features then pass through the decoder to reconstruct a noise-reduced map in the output, based on the training.

One advantage of Artificial Neural Networks (ANN) over the traditional methods is the ability to learn patterns. As seen in Fig. 3, a person walking on an RD map has certain patterns. Due to the difference in speed of legs and arms compared to the torso, a walking person appears to be a line-shaped object, compressing and stretching into a pattern that a human eye can identify and thus an ANN can learn it.

Since weights and biases of an ANN are randomized when generated, the way that training occurs varies from network to network. This fact could be beneficial by combining multiple CAE in parallel. This way, the system is less likely to get stuck in a local minimum while trying to converge during training. According to Yu et al. [14], combining multiple artificial neural networks can be used to achieve a higher pattern recognition since the sets of patterns wrongly recognized by different networks do not necessarily overlap. With this knowledge, we also have done experiments using multiple CAE during this research.

TABLE I
CAE architecture.

Layer	Type	Parameters	Output Data Size
conv1	Convolutional layer	32, 3x3, ReLU, Pad 0	128 x 128 x 32
pool1	Max Pooling	2x2	64 x 64 x 32
conv2	Convolutional layer	64, 3x3, ReLU, Pad 0	64 x 64 x 64
pool2	Max Pooling	2x2	32 x 32 x 64
conv3	Convolutional layer	128, 3x3, ReLU, Pad 0	32 x 32 x 128

(a) Encoder used in CAE.

Layer	Type	Parameters	Output Data Size
deconv1	Deconvolutional layer	128, 3x3, ReLU, Pad 0	32 x 32 x 128
up1	Up Sampling	2x2	64 x 64 x 128
deconv2	Deconvolutional layer	64, 3x3, ReLU, Pad 0	64 x 64 x 64
up2	Up Sampling	2x2	128 x 128 x 64
deconv3	Deconvolutional layer	1, 3x3, Sigmoid, Pad 0	128 x 128 x 1

(b) Decoder used in CAE.

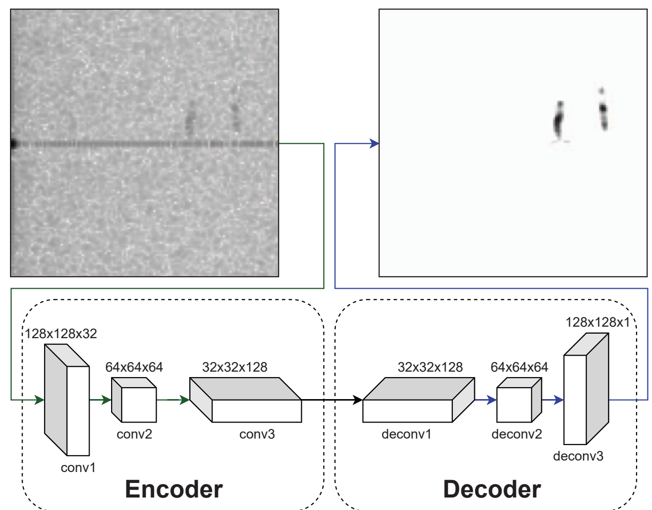


Fig. 2. Noisy Range-Doppler map passing through a 3 layer encoder and 3 layer decoder to obtain a cleaner image.

In addition to parallel networks, another technique we have attempted is an architecture that would include time series. Since RD maps are captured over time, multiple time-sequential frames could feed a network with the goal of trying to separate a moving object from just noise, since pure white noise changes from frame to frame. According to Zhao et al [15], time-series are an important class of temporal data objects and could be obtained through a chronological observation. In their work, they discuss a few ways to produce

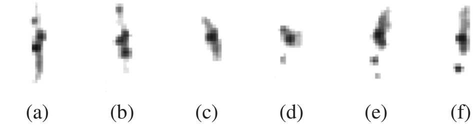
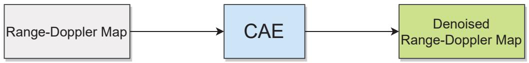
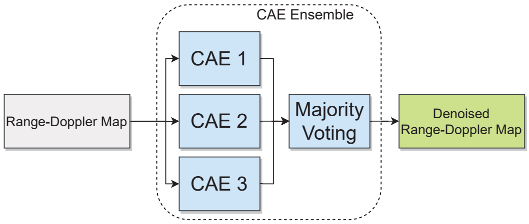


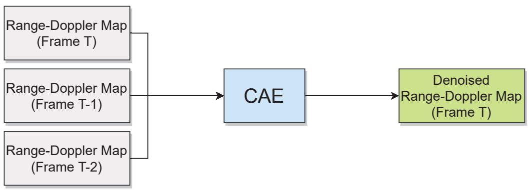
Fig. 3. Six frames of a person walking on an RD map, ordered by time - from frame (a) to (f).



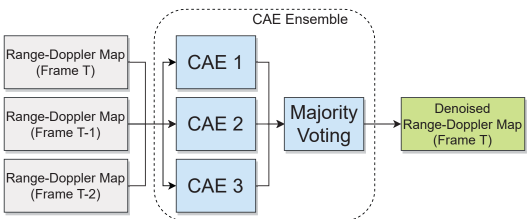
(a) 1 CAE - 1 frame



(b) 3 CAE - 1 frames



(c) 1 CAE - 3 frame



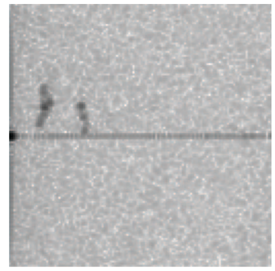
(d) 3 CAE - 3 frames

Fig. 4. Different configurations using CAE tested in this work.

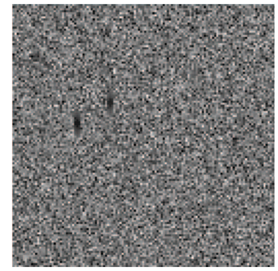
a time-series CNN.

Using the ideas of parallel networks and time-series, we proposed and tested 4 systems based on the CAE architecture: (i) a traditional 1 CAE that process one frame at a time (Fig. 4a); (ii) 3 CAE working in parallel, with different weights and biases, that process one frame at a time (Fig. 4b); (iii) 1 time-series CAE, that process the current frame and 2 previous frames (Fig. 4c); (iv) 3 time-series CAE working in parallel, processing three sequential frames (Fig. 4d). In order to provide a single result from the parallel networks, it is necessary to have a majority voting of their outputs. If at least two networks reconstruct an object in the same place, this object is preserved, otherwise considered a false positive and discarded.

While training CAE systems, we have noted that it was more efficient if the training was done step-by-step. First, a system would be trained with 1460 frames of algorithmically generated (synthetic) data with a PSNR value of 25 dB, followed by 1610 frames with a PSNR value of 10 dB, 1430 frames of 5 dB, and finally, 4550 frames ranging from 2 dB to 25 dB. If necessary, training using FMCW data would occur afterward. Peak Signal-to-Noise Ratio (PSNR) is a common way to measure image quality. PSNR is widely used for image and video quality measurement for its simplicity, and



(a) FMCW Radar



(b) Synthetic

Fig. 5. Two people walking on RD maps: the left being radar data and the right being algorithmically generated data.

therefore, intelligibility between researchers. A low PSNR means a high quantity of noise. A high PSNR value means the retrieved image is close to the original [16], [17]. During the development of this work, we have used PSNR as a measurement to either control the noise introduced algorithmically or to understand the amount of noise of a given RD map.

In order to analyze the noise-reduced results obtained with CAE, it was necessary to compare it with a well-known technique - such as CFAR or its variations. For this work, we used OS-CFAR (Order Statistic Constant False Alarm Rate), where the data is organized statistically before applying the CFAR filter. A Peak Detection (PD) algorithm is applied afterward for retrieving a less noisy RD map. For the algorithm employed in this work 10,000 Monte Carlo trials were used to calculate the false alarm rate.

For this work, we have used the number of correctly constructed objects as a form of measurement. To do so we applied a simple detector algorithm to calculate whether a denoised RD map reconstructed correctly the objects in their corresponding positions. We have used a foreground detector using three Gaussian modes in the mixture model. According to Piccardi [18], background detection (or background subtraction) is widely used to detect moving objects from stationary cameras. More about background detectors and Gaussian mixture model can be found in [18], [19].

III. Dataset

In order to be able to train and evaluate our systems, two databases were created: (i) A dataset using an FMCW radar from NXP, configured for near-field range, with bicycles and pedestrians; (ii) a dataset using algorithmically generated data (synthetic data), with the ability to control the amount of noise it would be introduced into the generated maps, therefore eliminating external problems and allowing the focus on CAE architecture. We have used the synthetic data for the first two experiments and FMCW radar data for the third experiment.

In Fig. 5 it is possible to see two people walking in Range-Doppler maps, however, the left image was extracted using FMCW radar information, and the right image was generated using an algorithm to simulate walking people.

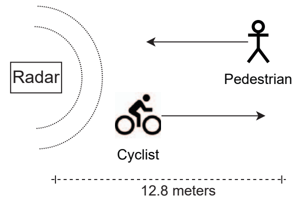


Fig. 6. An illustrative example of data collecting with a cyclist and a pedestrian.

TABLE II
EXPERIMENTS USING DIFFERENT ARCHITECTURES.

Method	Number of Convolutional Layers	Filter Size	Number of Parameters	Correctly Reconstructed Objects	Non-Existing Reconstructed Objects
CAE	5	3x3	5,478,851	97.5%	17.8%
CAE	4	3x3	1,349,059	97.7%	13.1%
CAE	3	2x2	140,835	96.3%	90.2%
CAE	3	3x3	316,355	98.0%	15.2%
CAE	3	3x3	79,843	97.6%	25.2%
CAE	3	3x3	1,259,395	**98.6%**	18.5%
CAE	3	4x4	562,083	97.4%	**10.7%**
CAE	3	5x5	878,019	97.0%	17.6%
CAE	2	3x3	58,051	95.5%	19.8%
OS-CFAR	-	-	-	38.2%	55.1%

A. FMCW Radar Data

It is impossible to research radars without collecting radar data. For this, we have used the 77 GHz TEF810X FMCW radar transceiver from NXP. Since our goal was to record bicycles and pedestrians walking from and to the radar, the configuration selected was near field, with a maximum velocity resolution of 24 km/h and a maximum distance of 12.8 meters. For this, 128 chirps, with a bandwidth of 1.8 GHz, were transmitting, each comprising 512 samples. After applying several 2D FFTs to the raw radar data, we obtained 3.782 Range-Doppler maps of 128 x 128 at 20 frames per second, in which 2007 frames were used for training and 1775 for evaluation. Then, to feed the CAE systems, the maps were converted to black and white images. Last, we labeled all objects manually.

During the data collection, we have used pedestrians and bicycles as moving objects. We set a maximum of two moving objects, in any arrangement, per frame. All objects moved either from or to the radar, so the relative radial velocity measured by the radar would be exact the relative velocity between them. Fig. 6 is an illustrative example of the data collection, in which a cyclist moves away from the radar and a pedestrian moves towards it. It is also important to note that people are not very reflective and their detected power from radars can be subtle, therefore sometimes, below the noise level.

We have collected two sets of data: (i) one with a stationary radar; and (ii) one with a moving radar (about 4km/h). A radar in movement introduces another noise to the RD maps since the relative moving ground could be misidentified as a moving object.

After collecting, separating and labeling all the FMCW radar data, we have created yet another set of data, injecting more Gaussian noise to all already collected data, decreasing the PSNR from an average of 9 dB to an average of 7 dB.

B. Synthetic Data

During the development of this work, we have noted the importance of making an algorithm capable of creating images that would mimic, at some level Range-Doppler maps. The advantages of doing so are various: to control the environment and its noise level; to generate a larger dataset; and to label objects automatically. Our goal was to describe, loosely, an RD map with up to six pedestrian-like objects walking, at various speeds, in which it was possible to control the Gaussian noise level (measured in PSNR) introduced. Following their collected data counterpart, we generated 18290 frames of 128 x 128 black and white images simulating RD maps, with PSNR varying from 2 to 25 dB, where 9050 frames were used for training and 9240 for evaluation. The synthetic data was generated using mathematical functions. We used a sinusoidal function to mimic the ever-changing velocity of the arms and legs regarding the torso, and an exponential function to simulate the power loss of a distant moving object.

IV. Experiments and Results

During this research, we made three sets of experiments: (i) using synthetic data, evaluate different architectures of convolutional autoencoders, regarding the number and depth of the layers; (ii) using synthetic data, study the performance of the four proposed CAE systems under various noise levels; (iii) using FMCW radar data, evaluate the performance of the same systems.

A. Evaluating Different CAE Architectures

For the first experiment, our goal was to determinate which combination of parameters (number of layers, filter size, depth) was necessary to produce a reliable and robust network that would not waste computational power. To perform this experiment, after pre-training each architecture (step-by-step like stated previously) they were once more trained just with 1250 frames of 5.5 dB (PSNR) of synthetic data. To evaluate our results, we used the number of correctly reconstructed objects (true positives) and the number of frames with reconstructed objects that did not exist in the original map (false positives).

The experiment results are shown in Table II. In this table, the number of convolution layers is regarding the encoder. For all architectures, the decoder size is equal to the encoder size. The number of parameters is related to the depth of each layer, filter size and the number of layers from both encoder and decoder. The Correctly Reconstructed Objects column is the percentage of objects that were correctly detected regarding the ground-truth map, in other words, true positives. In contrast, the Non-Existing Reconstructed Objects regards the percentage of frames where objects were detected but did not exist in the ground-truth map, therefore, it is the percentage of frames with a false positive. As seen in this table, all CAE architectures outperformed the traditional OS-CFAR with PD.

We have selected the CAE architecture with 3 convolutional layers in the encoder, 3 x 3 convolutional filter size, and

TABLE III
MEAN PERFORMANCE OF DISTINCT CONFIGURATIONS USING VARIOUS NOISE LEVELS.

Configuration	Correctly Reconstructed Objects	Non-Existing Reconstructed Objects
OS-CFAR	80.5 %	42.2%
1 CAE - 1 frame	93.9%	5.8%
3 CAE - 1 frame	93.6%	3.3%
1 CAE - 3 frame	**95%**	4.5%
3 CAE - 3 frame	94.6%	**1.6%**

316,355 trainable parameters as the one used during the other experiments because it was lighter than using 4 or 5 layers, and more robust than 2 layers or other configurations, therefore for our applications, it was believed to have the best cost-benefit. It is important to mention that, depending on the application, another configuration could have been chosen and would still outperform the traditional method.

B. Performance Over Various Noise Levels

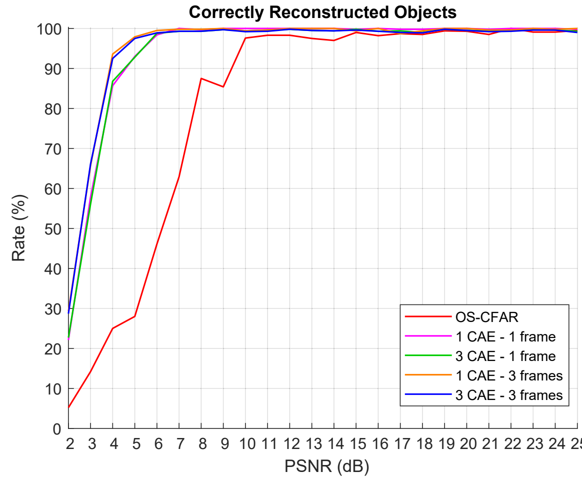
After we experimented with different CAE architectures, a new experiment was set. For this, we evaluated the four systems based on CAE over various noise levels: (i) 1 CAE processing one frame at a time; (ii) 3 CAE, working in parallel, processing one frame at a time; (iii) 1 time-series CAE, that processing 3 frames; (iv) 3 time-series CAE, working in parallel, processing 3 frames. Each system was trained with 9050 frames of the synthetic dataset and evaluated with 9240 frames - with a PSNR varying from 2 to 25 dB.

As the results are analyzed (shown in Fig. 7 and Table III), it can be inferred that any CAE system at any noise level outperforms OS-CFAR with PD. It is important to observe that systems using 3 frames (time-series) were slightly superior to those using only 1 frame regarding the number of correctly reconstructed objects; but for the number of non-existing objects, the systems using parallel networks performed more reliably. Although every system performs well, we believe that if extra computational power is available, the system using 3 parallel CAE and 3 sequential frames would be the most robust solution since it almost did not reconstruct non-existing objects (1.6% of frames) and it had the second-best correct reconstruction rate (94.6% of objects).

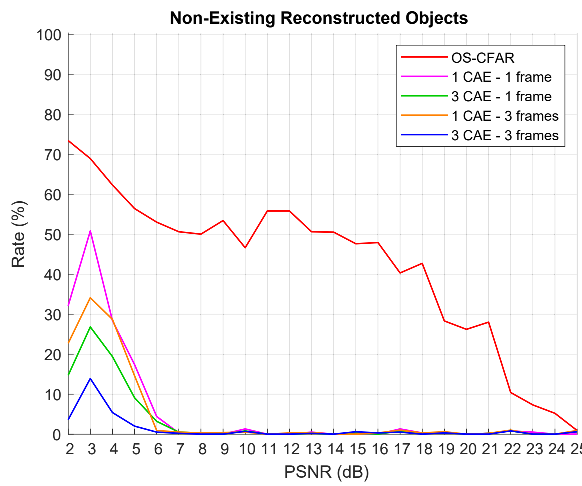
If compared with the traditional method, 3 CAE with 3 frames is 17.5% more efficient at correctly reconstructing objects and 25 times better at not reconstructing non-existing objects.

C. Performance Over FMCW Radar Data

In this last experiment, we have evaluated the same systems over FMCW radar data. For that, the data was separated into four groups: (i) stationary radar with measured noise; (ii) moving radar with measured noise; (iii) stationary radar with algorithmically elevated noise; (iv) moving radar with algorithmically elevated noise. For the training, 2007 frames were used. For evaluation, 890 frames of still radar and 885 frames of moving radar were used, a total of 1775 frames. The same amount of frames were used when using data with algorithmically elevated noise.



(a) Percentage of objects reconstructed in the right place.

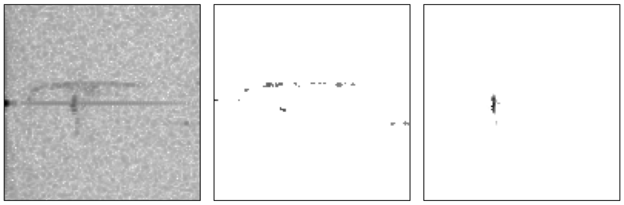


(b) Percentage of frames where non-existing objects were detected.

Fig. 7. Result of five distinct systems performing with various noise levels - from a PSNR of 2 dB to a PSNR of 25 dB.

The results can be analyzed in Table IV. It is possible to understand that, even though OS-CFAR is a good technique for certain conditions, it is not robust. In adverse conditions, it is difficult to distinguish between an object and noise. On the other hand, CAE systems presented greater robustness, thus being a viable solution. After analyzing all four sets of data, we would opt to use three parallel networks with three sequential frames because of its robustness: reconstructing 87% of the objects at the right place and reconstructing at least one non-existing objects in 20.1% of the frames. By comparison, OS-CFAR with PD reconstructed 49.2% of objects correctly and had 57.9% of frames with at least one false positive. This means an improvement of 76.8% and 188%, respectively, regarding the traditional method.

In Fig 8, an example of an RD map generated by a moving radar is shown alongside the comparison between the noise-reduced RD map from OS-CFAR with PD and a CAE system



(a) Original RD map (b) OS-CFAR with PD (c) 3 CAE and 3 frames

Fig. 8. Comparison between OS-CFAR with PD and a CAE system at reducing noise on a moving radar situation.

TABLE IV
PERFORMANCE OF VARIOUS CONFIGURATIONS USING FMCW RADAR DATA WITH BOTH MEASURED AND ALGORITHMICALLY ELEVATED NOISE.

Configuration	Correctly Reconstructed Objects	Non-Existing Reconstructed Objects
OS-CFAR	57.9%	**10%**
1 CAE - 1 frame	91.7%	13%
3 CAE - 1 frame	89.6%	**10%**
1 CAE - 3 frame	91.3%	13.7%
3 CAE - 3 frame	**91.9%**	11.5%

(a) Stationary radar with measured FMCW noise.

Configuration	Correctly Reconstructed Objects	Non-Existing Reconstructed Objects
OS-CFAR	54.7%	90.9%
1 CAE - 1 frame	**87.6%**	52.2%
3 CAE - 1 frame	83.4%	36.1%
1 CAE - 3 frame	87.1%	32.4%
3 CAE - 3 frame	84.8%	**24.6%**

(b) Moving radar with measured FMCW noise.

Configuration	Correctly Reconstructed Objects	Non-Existing Reconstructed Objects
OS-CFAR	45.4%	41.1%
1 CAE - 1 frame	**86.6%**	31.1%
3 CAE - 1 frame	85.2%	12.9%
1 CAE - 3 frame	85.3%	36.4%
3 CAE - 3 frame	85.3%	**11.4%**

(c) Stationary radar with algorithmically elevated noise.

Configuration	Correctly Reconstructed Objects	Non-Existing Reconstructed Objects
OS-CFAR	40.5%	90.2%
1 CAE - 1 frame	83.4%	62.6%
3 CAE - 1 frame	82.9%	35.9%
1 CAE - 3 frame	**85.7%**	71%
3 CAE - 3 frame	84.6%	**35.4%**

(d) Moving radar with algorithmically elevated noise

using 3 networks and 3 frames. While the CAE system can learn the patterns of a moving radar, the traditional method acts passively.

The different results obtained when working with FMCW radar data and synthetic data may be related to the fact that learning patterns with generated data is easier and more predictable. Besides, moving radars generate a different type of pattern on RD maps (which can also be seen in Fig. 8) that is distinct from Gaussian noise.

V. Conclusion

In this paper, we proposed the usage of Convolutional Autoencoders in various architectures and configurations to reduce noise levels in Range-Doppler maps due to the learning nature of neural networks in comparison with a traditional OS-CFAR method with a peak detector. While using synthetic data, the chosen CAE system correctly reconstructed 94.6% of the objects, compared with 80.5% from the traditional method - an improvement of 17.5%. The same system also had a false positive in 1.6% of the frames, in comparison with 42.2% of the frames while using the traditional method - a 25-fold improvement. While using FMCW radar data, the chosen CAE system showed an improvement of 76.8% to correctly reconstruct objects and 188% to not reconstruct non-existing objects regarding the OS-CFAR method.

This work is important because distant moving objects often are below the noise level, making it hard to retrieve information when using the traditional methods, but not for neural networks that can extract hidden features and learn patterns.

For future work, our goal is to use the benefits of a convolutional autoencoder to create a new tracking system using neural networks and how much computational power would possibly be saved during detecting and tracking when using ANN in comparison to the traditional way. Another application being pursued by us is to use autoencoders to generate more realistic synthetic data.

References

[1] M. Altmann, P. Ott, and C. Waldschmidt, "Deep Learning for Range-Doppler Map Single Frame Classifications of Cooking Processes," in *EuRAD*, 2018, pp. 194–197.

[2] F. Fioranelli, M. Ritchie, and H. Griffiths, "Classification of Unarmed/Armed Personnel Using the NetRAD Multistatic Radar for Micro-Doppler and Singular Value Decomposition Features," *IEEE GRSL*, vol. 12, no. 9, pp. 1933–1937, 2015.

[3] S. K. Joshi, S. V. Baumgartner, A. B. C. Silva, and G. Krieger, "Range-Doppler Based CFAR Ship Detection with Automatic Training Data Selection," *Remote Sensing*, vol. 11, no. 11, pp. 1–36, 2019.

[4] R. S. Narasimhan, A. Vengadarajan, and K. R. Ramakrishnan, "Design of connected component analysis based clustering of CFAR image in pulse doppler radars," in *IEEE Aerospace Conference Proceedings*, 2017, pp. 1–6.

[5] S. Blake, "OS-CFAR Theory for Multiple Targets and Nonuniform Clutter," *IEEE Trans. Aerosp. Electron. Syst*, vol. 24, no. 6, pp. 785–790, 1988.

[6] M. A. Richards, *Fundamentals of Radar Signal Processing*, 2005.

[7] W. M. Brown and R. J. Fredricks, "Range-Doppler Imaging with Motion through Resolution Cells," *IEEE Trans. Aerosp. Electron. Syst*, vol. 5, no. 1, pp. 98–102, 1969.

[8] M. S. Seyfioglu, A. M. Ozbayoglu, and S. Z. Gurbuz, "Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities," *IEEE Trans. Aerosp. Electron. Syst*, vol. 54, no. 4, pp. 1709–1723, 2018.

[9] N. Moreira, N. Leite, E. T. Pereira, E. C. Gurjao, and L. R. Veloso, "Deep Convolutional Autoencoder for EEG Noise Filtering," in *IEEE BIBM*, 2018, pp. 2605–2612.

[10] A. Azarang, H. E. Manoochehri, and N. Kehtarnavaz, "Convolutional Autoencoder-Based Multispectral Image Fusion," *IEEE Access*, vol. 7, pp. 35 673–35 683, 2019.

[11] Texas Instrument, "Traffic Monitoring Object Detection and Tracking Reference Design Using Single-Chip mmWave Radar," Tech. Rep. April, 2018.

[12] K. El-darymli, P. Mcguire, C. Moloney, and D. Power, "Target detection in synthetic aperture radar imagery : a state-of-the-art survey," *JARS*, vol. 7, pp. 1–36, 2013.

[13] M. Tschannen, O. Bachem, and M. Lucic, "Recent Advances in Autoencoder-Based Representation Learning," in *NeurIPS*, 2018, pp. 1–26.

[14] D. Yu, Q. Hu, and W. Bao, "Combining Multiple Neural Networks for Classification Based on Rough Set Reduction," *IEEE International Conference on Neural Networks and Signal Processing*, vol. 1, pp. 543–548, 2003.

[15] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, "Convolutional Neural Networks for Time Series Classification," *JSEE*, vol. 28, no. 1, pp. 162–169, 2017.

[16] A. T. Nasrabadi, M. A. Shirsavar, A. Ebrahimi, and M. Ghanbari, "Investigating the PSNR calculation methods for video sequences with source and channel distortions," *IEEE BMSB*, vol. 0, pp. 1–4, 2014.

[17] S. Qureshi, *Embedded Image Processing on the TMS320C6000TM DSP: Examples in Code Composer StudioTM and MATLAB*. Springer US, 2006.

[18] M. Piccardi, "Background subtraction techniques: a review," *IEEE SMC*, pp. 3099–3104, 2004.

[19] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," *IEEE CVPR*, vol. 2, pp. 246–252, 1999.