ISSN: 1983 7402 ITA, 28 a 29 SET 2021

Experimental Assessment of the Viability of a GAN Model for Radar Data Generation

Samuel Souza Alcântara Queiroz[1], Élcio Hideiti Shiguemori[2], Carlos Vinícius de Souza Augusto[1], Robson Porto de Souza[1], Roberto d'Amore[1] and Lester de Abreu Faria[1]

[1]Instituto Tecnológico de Aeronáutica (ITA), São José dos Campos/SP - Brazil

[2]Instituto de Estudos Avançados (IEAv), São José dos Campos/SP - Brazil

Abstract – **In the Defense industry, radar target simulations are a major point of technologic independence for countries, once allows creating and evaluating Radar Signatures of complex targets, as aircraft, ships and armored cars, without the need of measuring them. It is a possibility of generating enemy's signatures and defining the best approach to react and detect them. However, this kind of simulation is expensive, spend a lot of computational resources and demands a complex setup. This study proposes a proof of concept of a radar target simulation, based on a case study of automotive radars using Generative Adversarial Networks (GAN): an artificial intelligence technique that is used for generating realistic synthetic data. The achieved results show that it can generate realistic radar targets using low computational efforts, opening a new way to radar target simulations: AI-based simulations. Although being based on automotive targets, all results can be extrapolated to Defense scenarios.**

Keywords – **Generative Adversarial Network, FMCW Radar, Vehicle Safety.**

I. INTRODUCTION

Simulations are essential for all kind of complex studies, mainly for those in Defense and high added-value scenarios, where we do not have all elements, or that show to be very expensive to define parameters of operation. Through simulations it is possible to generate hypothetical scenarios, study them and define operation techniques, strategies, and approaches to overcome enemies in the war scenario. Among the different ways to simulate complex scenarios, one of them is known as *radar target simulation*, where a physical device, which consists of a transmitter and a receiver, generates a radar signal based on certain selected parameters. Such an approach to simulation is expensive and requires a lot of time and effort of those who are interested in running their simulations. There are only a few alternatives to this approach to this date, and none of them focuses on Defense area, in ostensive literature.

One of the little explored alternatives for radar signal simulations is *artificial intelligence*, more specifically the so-called Generative Adversarial Network [1] (or simply GAN). GAN is a network composed of Artificial Neural Networks that can learn how to generate realistic false (fake) data when it is fed with a dataset of examples. The GAN architecture has two neural networks: the generator and the discriminator. The generator is responsible for *generating* the "fake" data while the discriminator will try to distinguish between real or generated inputs. When the generator creates realistic data, the discriminator finds a high difficulty to distinguish between real and generated, and that is the point where every GAN needs to achieve in order to produce realistic results.

To this date, there are only a few numbers of works exploring GANs to radar signal generation. The existing approaches are for SAR (Synthetic Aperture Radar) specifically for remote sensing [2,3] and a simple approach of hidden object reflection [4]. This last one shows a certain fragility in the *discriminator*: the criteria used to make the final distinction between real and generated is only the human vision [4]. Therefore, authors conclude that the fragilities of the network are unknown [4].

The contribution of this paper rests on the viability of a GAN to generate realistic synthetic *range-doppler* radar data. For that, we used a case study of automotive scenarios, once it showed to be easier to find datasets that could be used to train the neural network, instead of Defense dataset, that are generally confidential and restricted by the governments. The results show that the case study and the GAN developed for this work its capable of generating *visually* realistic synthetic data, showing that artificial intelligence can be useful to simulation environments and even to replace expensive simulation setups, avoiding high computational resources.

The work [2] proposes a GAN that generates simulated satellite images. The objective is to supply a need for existing data for classification of SAR radar images. This work is divided into two parts: through an existing database, a Neural Network was created for classifying the objects found in the images, and through the same database, a GAN network was created to generate simulated images like the images present in the database. After these two phases, the generated images are added to the database, and the classification part is repeated to verify whether the increase in data implies an increase in classification accuracy. After doing the classification part, now with simulated images, the classification accuracy neither increased nor decreased. This implies that the generated images were not unrealistic enough to decrease accuracy, nor realistic enough to increase classification accuracy. The important point of this work [2] for the relevance of the project that will be carried out is the realism of the generated images and their discriminating network, and not the lack of increase in classification accuracy.

In [3] an approach focusing on the generation of simulated images is proposed. The problem that the work proposes to solve is the low precision of the existing simulations for SAR radar signals and images. Existing simulations are expensive, many of them requiring a very large computational effort and still delivering an inaccurate simulation. The proposal of the paper is to create an approach for simulations using a GAN. This network will learn using real data, and therefore learn to reproduce even more realistic data. The discriminator showed instabilities during training, sometimes causing the precision of the generated images to decrease. The problem was solved, but some noise from the radar still hindered the generation of more realistic images. The simulated images were tested in a CNN classifier and the results were compared with the classification of the real images. Classification accuracy on simulated images was lower.

In [4] a slightly different approach from the previous ones is presented. In this case it is not image simulation, but the radar signal itself. The cited work tries to solve a problem about the detection of hidden objects under clothes using a radar sensor, mainly for military applications. The work tries to create a proof of concept about the generation of simulated radar signals from a GAN network. A database was created from electromagnetic simulations using the Finite-Difference Time-Domain (FDTD) method. A backdrop was created: an object that is hidden behind some layers: a jacket, a shirt, and an object. Three classifications from the scenario were created: large object, small object, and no object. From the database, the GAN managed to capture the data distribution well and was able to generate simulated radar signals that are realistic to the human eye. The work presents a simple (not necessarily simplistic) proof-of-concept, with a scenario that is still not robust, as this is one of the first works to deal with the generation of radar signals from a GAN network.

In [5] the development of simulated images for snow sensor radar is shown. According to the paper, data collection over the years from remote sensing was done without an adequate classification, so the job of image classification for snow radar sensors becomes a difficult job to be carried out. One of the alternatives would be to generate synthetic data to fill this lack of classification. Data on the ice surface can be simulated through a physical simulation, but this type of simulation is computationally heavy (a very big effort) and is not feasible to produce a meaningful database for image classification. To accomplish this purpose, a CReSIS database (Center for Remote Sensing of Ice Sheets) with more than 2,000 images was used. The images generated by GAN had a good result compared to the real images, however the absence of some important input data caused some images to be generated with a lack of some ice layers, however, in general, the network was able to produce good results.

All these works show that Radar Signal Generation plays an important role in many areas, from Defense to Automotive, showing to be a major point of interest for specialists

II. THEORY

A. *Generative Adversarial Networks*

The GAN architecture (Fig. 1) consists in two neural networks: a generator and a discriminator. The generator learns how to create data using a random noise as a input, having a dataset (Shown as X_{train} in Fig. 1) as output [1]. The primary goal of the generator is fool the discriminator with the generated data. At the end of the process, if the generator can fool the discriminator, it means that the generated data its realistic enough to pass as real. In the training, the generator communicates with the discriminator to improve the data generation.

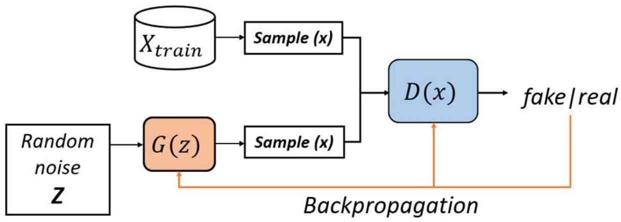

Fig. 1. GAN Architecture.

The main task of the discriminator is classifying the presented data as real or generated. During the training, the discriminator learns how to distinguish data from these two categories, calculates the generator loss and uses backpropagation to adjust the weights of the network. So, the discriminator helps the generator training improvement (Fig. 2). When the discriminator starts to classify the generated data as real, the goal was achieved. The behavior of a GAN is like a Turing Test: to be considered a "thinking" network, the generator must pass as real to the discriminator.

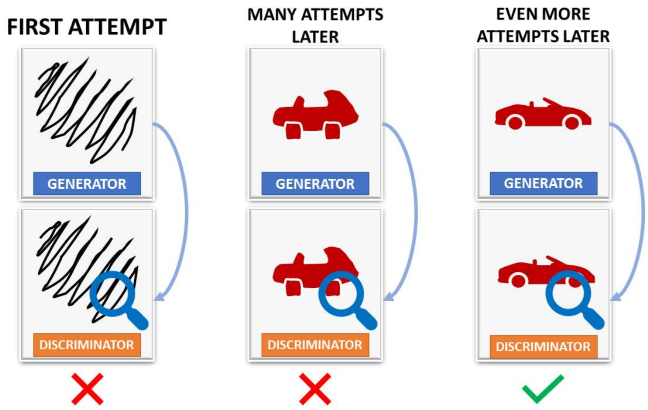

Fig. 2. GAN training process.

This architecture can be used to generate any kinds of fake data: texts, audios, music, videos, images, and more [6].

B. *Frequency Modulated Continuous Wave Radar (FMCW Radar)*

The FMCW radar is a type of a radar sensor that can change by being modulated, differently from the simple continuous wave radar (CW) [7]. The CW radar has limits: it can't measure the range of target. This problem is caused by the lack of the time mark, which allows the system to do its cycle of transmist, receive and convert this process into range [7]. The time reference for the measurement of objects distance can be generated by using the frequency modulation of the transmitted signal [7], as shown in Fig.3.

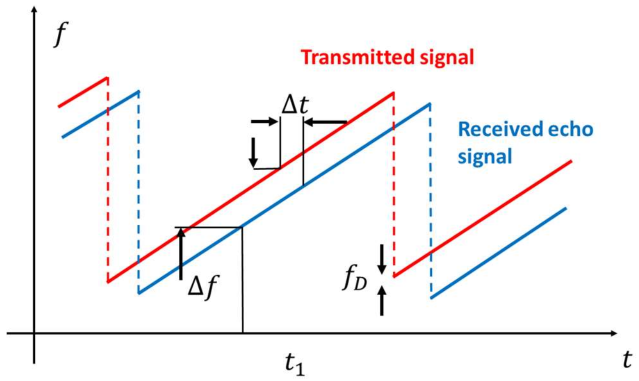

Fig. 3. FMCW transmit and receive cycle.

The distance R to the reflecting object can be calculated by the following equation (1):

$$R = \frac{c_0\,|\Delta t|}{2} = \frac{c_0\,|\Delta f|}{2\left(\frac{d(f)}{d(t)}\right)} \qquad (1)$$

Where:

- c_0 is the speed of light.
- Δt is the time delay (s).
- Δf is the measured frequency difference (Hz).
- R is the distance between the antenna and the reflecting object (m).
- $\frac{d(f)}{d(t)}$ is the frequency shift per unit of time.

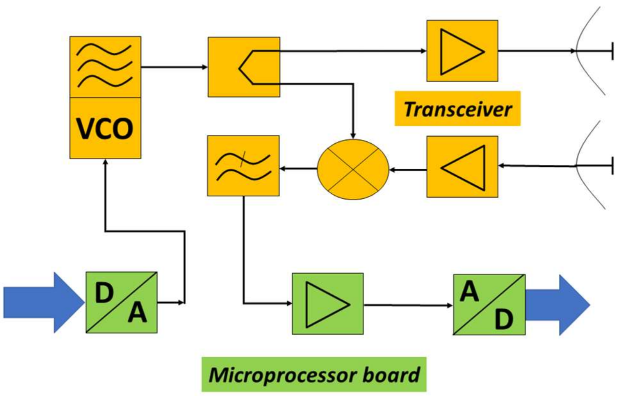

Fig. 4. Radar block diagram

The radar is basically the transceiver and a control unit with a microprocessor (Fig. 4). The transceiver includes a transmitter and receiver antennas. A high frequency is generated by an oscillator that controls voltage. This oscillator feeds the transmitting antenna or amplifies its power [7]. The mixer is fed by a part of the high frequency, and then converts the received echo signal in the baseband [7]. The microprocessor controls the transceiver, converts the signal to the digital format, usually with a USB cable [7]. The control voltage is provided to the frequency control by using a digital-to-analog converter, then, the mixer output voltage is converted to the digital format.

III. Methodology

A. CARRADA Dataset

The CARRADA Dataset [8] is a database containing about 80 gigabytes of automotive radar measurements together with a camera. From these measurements, an FMCW (Frequency Modulated Continuous Wave) radar database was created along with its post-processing. About three objects were used: cars, bicycles, and cyclists. The results of this collection were separated as follows: camera images, range angle sequences and range doppler sequences. Each collection of each one of the frames has an identifier that synchronizes the images and radar sequences (Fig. 5). The collected data was saved in 2D arrays in a format called NumPy array, which can be read using a python algorithm. In addition, each frame of collected data was processed through a Convolutional Neural Network (CNN) that detects the present objects in the scene, maps them, performs semantic segmentation, and marks the objects in the images. This dataset is of great importance for artificial intelligence aimed at automotive radars, once databases available for FMCW radars are practically non-existent, or closed to the public. The database is available for using via github.

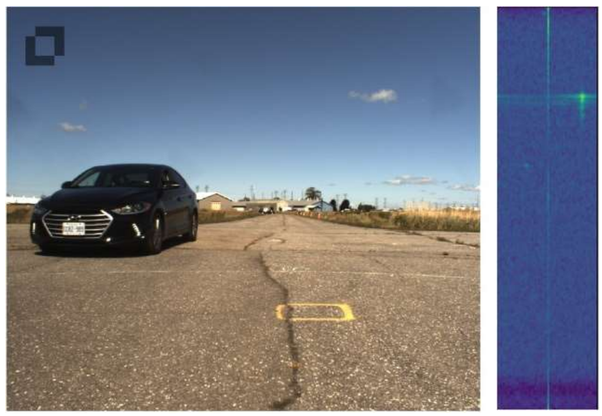

Fig. 5. Camera and range-doppler raw data representation.

In the present work, not all measurements of the dataset were used, but only the listed below, which involves about 6.386 radar measurements:

- Range-doppler raw data representations; and
- Empty measurements and car measurements.

The data needs to be processed previously to fit into the neural network. A resize was needed to pass the data through the network. The representations were resized from 256 x 64 to 64 x 64 (Fig. 6).

ISSN: 1983 7402

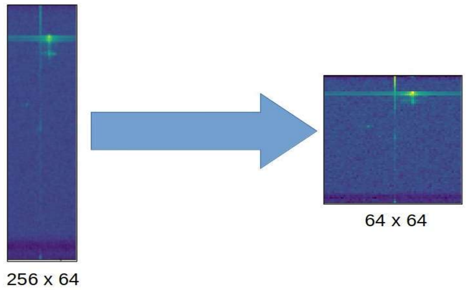

Fig. 6. Resizing data.

To train a model that learns the data representation and generates realistic synthetic data, the GAN needs a large amount of data [9]. The data used in this experiment is not enough to train a realistic model. But, according to [9] a technique of *data augmentation* can be used to enlarge the dataset. Data augmentation basically is a way to augment the dataset applying different kinds of transformations to the data and adding those transformed data as new data. For example, a new color can be applied to an image, and that new colored image as a new data. Every type of data has its adequate transformations that can be applied. But to a range-doppler representation, a new color cannot be applied, so, to this experiment, a series of *gaussian noise* were applied to the data (Fig. 7) and added as new representations. Therefore, the dataset achieved 57,474 measurements, what is enough to train a good generative model.

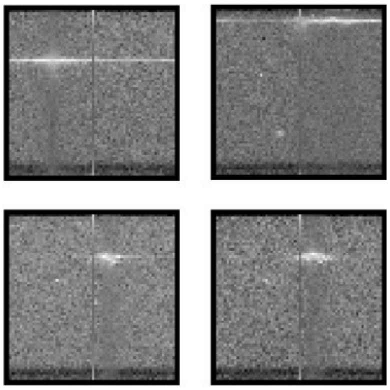

Fig. 7. Data with normalization and gaussian noise.

B. Deep Convolution Generative Adversarial Network

The selected GAN architecture is known as Deep Convolutional GAN (DCGAN). This type of GAN replaces the fully connected layers in the generator with upsampling convolutional layers [10]. Mainly, there are three pillars of the DCGAN architecture [10]:

1. Replace pooling operations by downsampling convolutions.
2. Remove fully connected layers after the convolutions.
3. Use Batch Normalization to help the gradient flow.

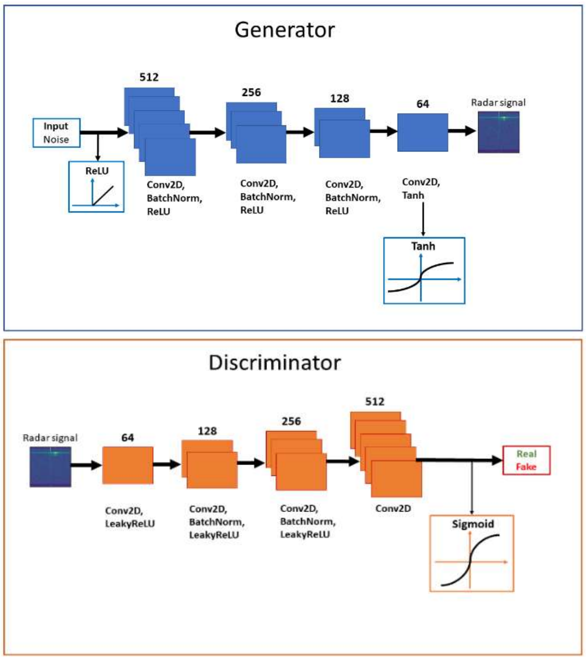

Fig. 8. Architecture used in the experiment.

The DCGAN uses the unsupervised training, that means that the data is not labeled. The DCGAN architecture also uses the ReLU activation function in the generator for all layers (except the output), and in the output, it uses the Tanh activation function. In the discriminator it uses LeakyReLU for all layers and the Sigmoid function in the classification.

IV. Experiments and Results

This section presents the setup used in the experiment, the metrics to evaluate GAN model, a qualitative analysis of the generated data with visual comparation between real and generated data and quantitative results using the Geometry Score [11] to compare the data distribution of the real and generated data.

A. Experiment setup

To accomplish the experiment, a configuration setup was built:

- Google Colab with GPU to train the model.
- 30 epochs of training.
- Batch size of 128 samples.
- A learning rate of 0.001.

B. Finding a metric

Finding a metric to evaluate the accuracy of a model is not a trivial task, specially when it comes to GAN models and it comes more complicated with radar signals. There are a few and unsufficient metrics that can evaluate an artificial generated radar signal. In the reference [4] authors use a visual comparation between the real and the generated data. The work [2] uses a CNN classifier to evaluate the model, but this method is inadequade to the present experiment: there are no label attached to the data itself, therefore impossible to make a classification model. Looking at these problems of a good metric to evaluate the model, two solutions came up:

1. A qualitative analysis by visual comparation, as used in [4].
2. A quantitative analysis, the *geometry score* [11].

C. Qualitative analysis

In order to compare the real and the generated data, 4 representations of each one of them were randomly selected. The results are shown in the Figure 9.

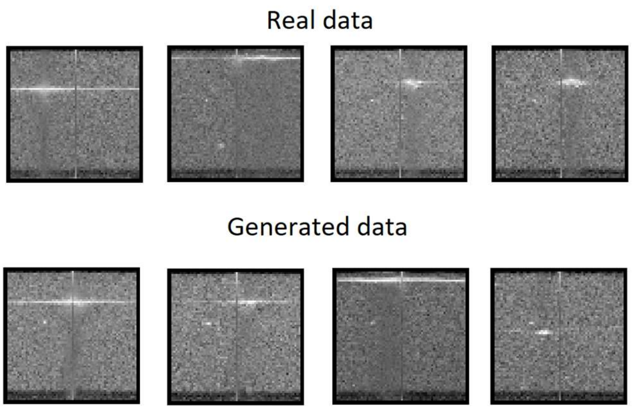

Fig. 9. Real data vs generated data.

The data used to train the model is a *non-processed* data, therefore the *raw data* of the radar. The generator also generates the *raw data* from the range-doppler representation.

The response shown in the data is the *range speed pattern*, where the *x* axis is the velocity, and the *y* axis the range (distance from the target). A closer look a generated image is shown in Figure 10.

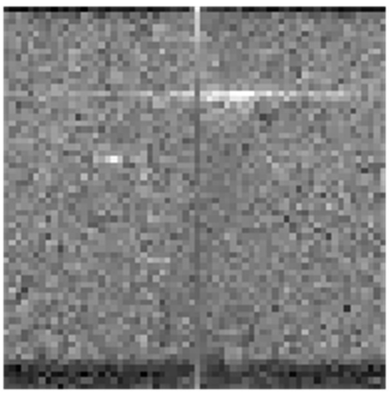

Fig. 10. Artificially generated range-doppler representation.

As shown in the Figure 10, the model was clearly able to generate a visually convincing range-doppler response to a target. It is possible to see the target and even other object in the scene. The model was able to generate a target in a range speed pattern visually similar to the patterns in the real data. With the output data its possible to see that:

- There is only one target in the scene.
- The target is moving relatively to the radar: the velocity (*x* axis) is different from zero (the middle is zero).
- The data needs to be processed: that "line" in the axis *x* wasn't supose the be there, the car is a single point object.
- The response generated by the model is visually similar to the real: it could fool the human vision analysis.

D. Quantitative results

In order to complement the visual analysis of the generated data, the metric *geometry score* [11] was selected in order to evaluate the precision of the model.

The geometry score estimates the quality and diversity of the genereted data when compared to real data. It goes through the topology of the underlying manifold of the generated data samples to check how different the samples are from the topology of the real data [11].

This method is based in the probabilistic understanding when two datasets X_1 and X_2 are given, as shown in the equations (2) and (3):

$$GeomScore(X_1, X_2) \tag{2}$$

$$\sum_{i=0}^{i_{\max-1}} \left(MRLT(i,k,X_1) - MRLT(i,k,X_2)\right)^2 \tag{3}$$

Where:

- MRLT is the *Mean Relative Living Times*.
- X_1 and X_2 are the datasets to be compared.
- i is the number of suffices.
- k is the number of dimensional holes in the underlying manifold of X. In this case k = 1 is used.

To measure a topological structure, it is important to map each point of the dataset. In these points a Čech Complex [12] can be used to place balls at each point of the dataset. Mapping the dataset, some holes appear, some of them "lives" more, and others have a short period of life [12]. The *Relative Living Times* (RLT) is calculated by taking the ratio of the total amount of the time that a number of holes appears and the maximum value of time [12].

In order to calculate the Geometry Score, it is necessary to calculate the *mean* of the RLT (or simply MRLT). In the equation (2) The MRLT from the *real* dataset is substracted from the MRLT of the *generated* dataset. The values are summed for each suffice, and the sum is the *Geometry Score*. This method compares the data probability distribution between two datasets and how far their topology are from each other.

In Figure 11 it is possible to see the differences between the real and the generated dataset. The MRLT for each number of one-dimensional hole (i) is presented. As shown in the Figure 11 the data distribution of the generated dataset have only a few differences between the real dataset. The MRLT of the generated data is something about 0.61 with the number of *zero* one-dimensional holes, and the real dataset has a MRLT of 0.9 with the number of *zero* one-dimensional holes in the its topology.

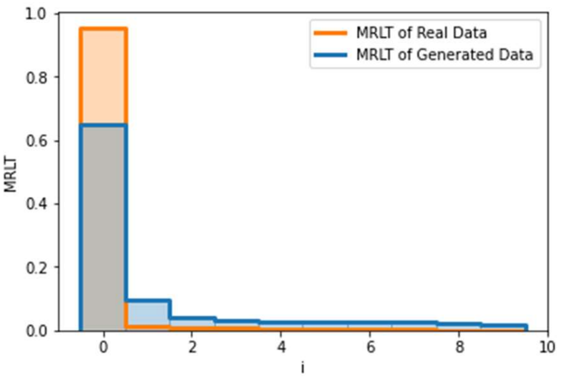

Fig. 11. MRLT of real and generated data.

With the MRLT, now it is possible to calculate the *Geometry Score*. The work [11] presents results using the MNIST Dataset hand written digits. The scores for each categorie varies from 0.47 to 22.8, being the lower value the best score. The closer to zero the value of the score, the more similar to the real topology. The score calculated for this experiment was **0.103.**

With these results in mind, its possible to conclude that:

1. The GAN model was able to generate data with a topology similar to the real one; and
2. The parcial results of this study shows that AI can generate realistic data and improve high level simulations.

The *Geometry Score* does not evalute visual aspects of the generated data and nothing sugests that if the topology of data is near to real, then the image should be similar to the real ones [12]. That's why in this study two methods were used, a qualitative, to evalute the visual aspects, and a quantitative. Those two combined can give a good evaluation for the generative model.

It is important to say that these results are still preliminar, and therefore not conclusive. Despite the fact that the GAN model showed potential to create realistic radar data, the scenarios used in this experiment were simple, with just one object, few background noise and the unsupervised learning was used. New studies are needed to evaluate how a GAN model would behave in the presence of more complex data, with multiple objects in the scene, noise and very different targets.

VI. Conclusions

This article assessed the viability of Generative Adversarial Networks to generate radar range-doppler representations. The results shows the the generative model was able to create visually realistic radar data that is able to fool the human eye and topological similar to the real data. Despite the fact that this is a initial study, the model showed a high potential to generate realistic radar data.

The GAN architecture can be used in radar simulations. The model created can be saved and used elsewhere with low computation costs when compared to a normal simulation setup. It takes a lot of computational effort to *train* the model, but once the model is trained, it can be exported and used with lower computation resources. This opens a way to improve radar simulations with artificial intelligence. Hopefully, new models will be developed with more complex data and will be able to generate scenarios with more targets, noise and different objects.

In future works, a Conditional GAN (CGAN) will be explored: this one can generate data according to a label. The dataset used is labeled with different categories, so the model learns to generate a data with a category input. For example: The user could pass as input "car" and the model would return a range-doppler represention of a car.

References

[1] 6 TensorFlow Core, "Deep Convolutional Adversarial Networks", Fev 2021, Available in: <https://www.tensorflow.org/tutorials/generative/dcgan>. Access in: 8/07/2021.

[2] 1 D. Marmanis, W. Yao, F. Adam, M. Datcu, P. Reinartz, K. Schindler, J. D. Wegner, U. Stilla, "Artificial generation of big data for improving image classification: a generative adversarial network approach on sar data", Nov 2017, Available in: <https://arxiv.org/abs/1711.02010>. Access in: 30/04/2021.

[3] 2 J. Guo, B. Lei, C. Ding, Y. Zhang, "Synthetic aperture radar image synthesis by using generative adversarial nets", IEE Remote Geoscience and Remote Sensing Letters, 2017.

[4] 3 T. Truong, S. Yanushkevich, "Generative adversarial network for radar signal generation", Aug 2017, Available in: <https://arxiv.org/abs/2008.03346>. Access in: 30/04/2021.

[5] 5 M. Rahnemoonfar, M. Yari, J. Paden, "Radar sensor simulation with generative adversarial network", Out 2020, Available in: <https://www.researchgate.net/publication/344818488>. Access in: 30/04/2021.

[6] 7 I. J. Goodfellow et al, "Generative Adversarial Nets", Jun 2014, Available in: <https://arxiv.org/abs/1406.2661>. Access in: 08/07/2021.

[7] 8 C. Wolf, "Frequency Modulated Continuous Wave Radar (FMCW)", available in <https://www.radartutorial.eu/02.basics/rp08.pt.html>. Access in 09/07/2021.

[8] 4 A. Ouaknine, A. Newson, J. Rebut, F. Tupin, P. Pérez, "Carrada dataset: camera and automotive radar with range-angle-doppler annotations", Fev 2021, Available in: <https://arxiv.org/abs/2005.01456>. Access in: 30/04/2021.

[9] 9 T. Karras et al, "Training Generative Adversarial Networks with Limited Data", Jun 2020, available in <https://arxiv.org/abs/2006.06676>. Access in 05/05/2021.

[10] 10 C. Shorten, "Deeper into DCGANs", Feb 2019, available in <https://towardsdatascience.com/deeper-into-dcgans-2556dbd0baac>. Access in 09/07/2021.

[11] 11 V. Khrulkov, I. Oseledets. "Geometry Score: A Method for Comparing Generative Adversarial Networks", Jun 2018, available in <https://arxiv.org/abs/1802.02664>. Access in 09/07/2021.

[12] 12 G. Vargas. "Geometrical Evaluation of Generative Adversarial Networks", Jul 2019, available in < https://quantdare.com/geometrical-properties-of-generative-adversarial-networks/>. Access in 09/09/2021.