Exploiting GAN Capacity to Generate Synthetic Automotive Radar Data

Mauren Louise S. C. de Andrade[1], Matheus Velloso Nogueira[2], Eduardo Candioto Fidelis[2],
Luiz Henrique Aguiar Campos[2], Pietro Lo Presti Campos[2], Torsten Schön[3] and Lester de Abreu Faria[2]
[1]*Universidade Tecnologica Federal do Parana, Ponta Grossa, Brazil*
[2]*Centro Universitario Facens, Sorocaba, Brazil*
[3]*AImotion Bavaria, Technische Hochschule Ingolstadt, Ingolstadt, Germany*

Keywords: Radar Application, Generative Adversarial Network, Ground-Based Radar Dataset, Synthetic Automotive Radar Data.

Abstract: In this paper, we evaluate the training of GAN for synthetic RAD image generation for four objects reflected by Frequency Modulated Continuous Wave radar: car, motorcycle, pedestrian and truck. This evaluation adds a new possibility for data augmentation when radar data labeling available is not enough. The results show that, yes, the GAN generated RAD images well, even when a specific class of the object is necessary. We also compared the scores of three GAN architectures, GAN Vanilla, CGAN, and DCGAN, in RAD synthetic imaging generation. We show that the generator can produce RAD images well enough with the results analyzed.

1 INTRODUCTION

Avoid and preventing car accidents are the main objective when the automobile industry uses sensors like LiDAR, digital cameras, and radar. So, it is fundamental to guarantee that kind of technologies will return the correct information about the vehicles and what is around them. For this, many researchers have dedicated their effort to studying and creating a more confident use of those sensors (Schuler et al., 2008), (Liu et al., 2017), (Eder et al., 2019), (Deep, 2020), (Lee et al., 2020), (Ngo et al., 2021). Unfortunately, the high cost of LiDAR and the problems in capturing images in adverse weather conditions limit its use.

On the other hand, adding radar technologies such as Frequency Modulated Continuous Wave (FMCW) in automobile devices can be an efficient and cost-effective choice even in low light conditions and bad weather situations. In this case, the purpose of the electromagnetic waves produced by the radar is the detection of the position, velocity, and characteristics of targets by the radar. With this aim, the radar transmits electromagnetic waves and receives the echo returned after being reflected, detecting both the presence of targets and their spatial location (Rahman, 2019), (Iberle et al., 2019).

In a nutshell, the electromagnetic waves emitted by the radar are used to detect objects through signal strength captured by the receivers. When these waves hit an object, they are reflected in different directions. Just a tiny part of the emitted energy returns to the radar, susceptible to noise such as corruption, thermal noise, electromagnetic interference, atmospheric noise, and electronic countermeasures (Rahman, 2019), (Danielsson, 2010). Then, the raw radar data generated can be represented as 3D tensors that illustrate the range, angle, and velocity called the Range-Azimuth-Doppler (RAD) spectrum. Figure 1 shows a Range-Azimuth-Doppler (RAD) spectrum sample and the targets as well as the camera image of the scene.

However, interpreting the radar data is not easy task, mainly due to the scarcity of validated data available (Wald and Weinmann, 2019), (de Oliveira and Bekooij, 2020). Besides, the data collected and generated by the radars must be validated and calibrated in an experimental field to ensure the reliability of systems, specifically when related to safety.

To study, analyze and test the FMCW radar's data, the researchers use simulators, such as Carla and Radar Toolbox by Matlab, among others (Dosovitskiy et al., 2017), (MathWorks, 2022). In this case, it is possible to simulate the radar at different levels of abstraction through mathematical and probabilistic models provided by the simulators or through the insertion of real data acquired by the radar. In this way,

C. de Andrade, M., Nogueira, M., Fidelis, E., Campos, L., Campos, P., Schön, T. and de Abreu Faria, L.
Exploiting GAN Capacity to Generate Synthetic Automotive Radar Data.
DOI: 10.5220/0011672400003417
In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 4: VISAPP*, pages 262-271
ISBN: 978-989-758-634-7; ISSN: 2184-4321

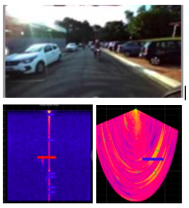

Figure 1: Range-Azimuth-Doppler spectrum and it's targets. Range-Azimuth on the lower left and Range-Doppler on the lower right.

it is possible to analyze the experimental data more realistically to ensure that the signal radar use and interpretation are more reliable in different contexts.

However, both the real data generated by the radar and those produced artificially must contain a significant number of samples in different scenarios. In this case, synthetic data makes it possible to increase the number of samples available. So far, two main approaches have been used to generate synthetic radar data for use by simulators: models based on ray-tracing and based on artificial neural networks.

The ray-tracing method is based on ray optics that solve the Maxwell equation at high frequency, modeling the propagation with estimates of path loss, arrival/departure angle, and time delays. It is a computer program and is a numerical method that solving the Maxwell equation (Yun and Iskander, 2015). However, the complexity of the calculations is a negative aspect of applying this methodology in the generation of synthetic radar data since it requires high computational power, consequently, difficult its application on a large scale (Magosi et al., 2022).

On the other hand, methods based on neural networks are designed to learn from the representations of high-level features of the input data, which are used to make intelligent decisions in different subjects such as engineering, medicine, environmental, agriculture, technology, climate, business, arts, and nanotechnology, etc (Abiodun et al., 2018). Some standard deep learning networks include deep convolutional neural networks (DCNNs), recurrent neural networks (RNNs), generative adversarial network (GAN) and automatic encoders, among others (Abdu et al., 2021a).

In the specific case of synthetic radar data generation, architectures based on autoencoder neural networks and generative adversarial networks have been successfully applied to overcome the disadvantages of traditional ray-tracing approaches (not deep learning) (Song et al., 2019), (de Oliveira and Bekooij, 2020). However, the application to generate synthetic data related to RAD images is scarce. In general, the algorithms of these classes present problems related to the availability of labeled data that compose the model to be trained (Jiao and Zhao, 2019).

GAN is from the family of artificial neural networks that were developed to generate data with characteristics similar to the real input data. The basic idea is the combination of two networks that train together: the generator, which uses a vector of random values as input and generates data with the same structure as the training data; and the discriminator, which uses dataset containing observations from both training and generator-generated data, to then classify the observations as ”real” or ”generated” (Radford et al., 2016).

So, motivated by the challenge of generating synthetic radar data, this paper will use the adversarial training concept to create new RAD images from an image database with four types of objects generated through the captured FMWC radar. For this purpose, we will use a labeled image database using images of the scene provided by a camera system as guide. We will apply and examine three generative adversarial network approaches to generate synthetic radar data (GAN (Goodfellow et al., 2014), CGAN (Mirza and Osindero, 2014a) and DCGAN (Radford et al., 2015)). Even so, as far as we know, there is still no GAN assessment in RAD imaging generation. Therefore, this article adds RAD imaging generation by GAN as a new perspective for using synthetic data as data augmentation for RAD image datasets for posterior application on simulators.

In this article, besides to comparing the best performance in generating RAD images between the GAN, CGAN, and DCGAN networks, we analyzed the applicability of the generated data for data augmentation. To do so, we evaluated the ability of the generator to produce compelling images enough to make it difficult for the discriminator task to distinguish between authentic and false images. Furthermore, we evaluated the influence exerted on the values of the scores obtained by training of generator and discriminator when specific parameters are changed.

The main contributions of this article are:

- evaluation of the best performance between three GAN architectures (traditional GAN, CGAN, and DCGAN) in generating synthetic RAD images;

- assessment of the ability of the generator to produce images good enough to make it difficult for the discriminator to distinguish between ”real” and ”fake” images;
- analyze of the influence of values of the scores obtained by the GAN when specific parameters are changed;
- analyze the applicability of the synthetic RAD image generated for data augmentation.

The paper is organized as follows. In Section 2, works related to the generation of synthetic radar data are presented. In Section 3, the generative model network is briefly presented. In Section 4, the experimental setup used are described. In Section 5 the results achieved by training GAN methods are presented in order to validate the generated databases. Finally, conclusions are provided in Section 6.

2 RELATED WORK

Traditionally, radar is used to identify objects around itself. Its use in defense, air traffic control, maritime control, and air industries is known for its ability to locate objects over long distances (Javadi and Farina, 2020). However, the applicability of radar data has been expanding as the cost of acquiring the radar becomes more affordable. Unlike the initially idealized applications, the radar can be positioned directing towards identifying smaller objects closer to the radar, such as vehicles and people on public roads, in open or closed environments.

Therefore, an adequate analysis of the radar data is necessary, especially regarding vehicle safety. In this case, the RAD spectrum will present small nuances that will allow the correct identification and subsequent classification of the reflected object. In general, for a classification to be considered promising, it must be used in large volumes of labeled data, which is not easy to obtain as described above.

This section will briefly review existing work on synthetic radar data generation and its applications. The Table 1 provides a summary of the works. Note that the papers in Table 1 were selected for their particularity on generated synthetic radar data images for different contexts. Based on the literature review results, we then justify why our work proposes the GAN networks to provide data augmentation of RAD images.

We can see in Table 1 that most of the works in the literature apply GAN networks to generate synthetic radar data instead the traditional data augmentation (Skeika et al., 2020) . One of the reasons is the traditional data argumentation that involves rotation, translation, noise injections, alteration in the color, brightness, contrast, and cropping, among others, is impossible for range-azimuth and range-Doppler images (Skeika et al., 2020), (Kern and Waldschmidt, 2022). For example, (Alnujaim et al., 2021), (Qu et al., 2021), (Vishwakarma et al., 2021), (Erol et al., 2019) used different adversarial generative network architectures for data augmentation of range-Doppler images in human gestures classification. In these works, the main idea is to guarantee a large enough dataset to classify human gestures and activities correctly. When classified correctly, the signature of human motions can be used as an auxiliary in defense, surveillance, and health care. Variations in architecture were between works by (Alnujaim et al., 2021), (Erol et al., 2019), (Rahman et al., 2022) using a conditional GAN framework, (Rahman et al., 2021), (Doherty et al., 2019), (Vishwakarma et al., 2021) using a traditional GAN and (Radford et al., 2015) using DCGAN.

The works closest to ours are (Wheeler et al., 2017) and, (de Oliveira and Bekooij, 2020) since their objective was to conduct experiments to increase vehicle safety. In Wheeler, T. A. et al. (Wheeler et al., 2017), for example, they generate synthetic radar data for use in automotive simulations. The authors presented an approach to generating synthetic radar data, an architecture based on DCGAN. For this, the Variational Autoencoder architecture was used to synthetically simulate the reading of a radar sensor. In the application, the network receives two inputs: (a) a 3D tensor representing the Range-Azimuth map of the radar and (b) a list of objects detected by the sensor, containing each identified object and its characteristics. The model was able to generate synthetic radar data with low complexity of scenario (with few objects).

In this way, to create synthetic Range-Doppler maps of FMCW radars for short range and with moving objects (pedestrians and cyclists), de Oliveira and Bekooij (de Oliveira and Bekooij, 2020) made use of GAN and Deep Convolutional Autoencoders. The data consists of multiple Range-Doppler (RD) spectra with and without targets, collected with an FMCW radar. The authors place both frames in sequence and add noise to the synthetic RD maps, groups of five series. They are training a system detector using synthetic data and evaluating it with real data from the FMCW radar, with good results.

Although the examples of GAN in the generation of synthetic radar data described are promising, so far, we are unaware of the use of the adversarial training concept in the creation of new RAD spectrum images

Table 1: A brief summary of the existing works on synthetic radar data generation.

Author	Application	Methodology
Alnujaim, I., et al. (Alnujaim et al., 2021)	Human body	CGAN
Qu, L., et al. (Qu et al., 2021)	Human data	WGAN-GP
Vishwakarma, S., et al (Vishwakarma et al., 2021)	Human data	GAN
Erol, B., et al. (Erol et al., 2019)	Human gesture	ACGAN
Wheeler, et al. (Wheeler et al., 2017)	Radar data	DCGAN
M.M. Rahman, et al. (Rahman et al., 2021)	Human data	GAN
M.M. Rahman, et al. (Rahman et al., 2022)	Human data	CGAN
Doherty, H. G., et al. (Doherty et al., 2019)	Human data	GAN+AAE
Oliveira et al.(de Oliveira and Bekooij, 2020)	Pedestrian and cyclists	GAN and DCA

for the four objects proposed here (car, pedestrian, truck and motorcycle).

3 GENERATIVE ADVERSARIAL NETWORK

The Generative Adversarial Network (Goodfellow et al., 2014), popularly known only by its acronym GANs, was presented as an alternative framework to train generative models. They are made up of two network blocks, a generator, and a discriminator. The generator usually takes random noise as its input and processes it to produce output samples that look similar to the data distribution (e.g., false images). In contrast, the discriminator tries to compare the difference between the actual data samples and those produced by the generator (Abdu et al., 2021b).

The main idea of the framework is to estimate generative models through an adversarial process, in which both models are trained simultaneously: generative G that captures the data distribution and discriminative D that estimates the probability that a sample came from the training data instead of G.

The training procedure for G is to maximize the probability that D will make an error. This framework corresponds to a minimax game for two players. In the arbitrary space of the functions G and D, there is a single solution, with G retrieving the distribution of the training data and D equal to $\frac{1}{2}$ everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation (Lekic and Babic, 2019).

3.1 Loss Function and Scores

In adversarial generative networks, the generator produces images that the discriminator evaluates, whether they are real or false. To maximize the probability of these generated images being classified as real by the discriminator, minimizing the negative log likelihood functions is necessary. To do so, consider the output Y as the probability of the discriminator. That is, Y is the probability of the generated image being ”real,” and $1-Y$ is the probability of the image being ”false.” A loss function is calculated to penalize wrong forecasts. In this case, if the probability associated with the ”real” class is equal to 1.0, then the cost will be zero. Likewise, if the probability associated with the class is low, close to 0.0, then its cost will be high. Therefore, the definition of the cost function for the generator is (Goodfellow et al., 2014):

$$lossG = -mean(log(Y_{Gen})), \tag{1}$$

where Y_{Gen} will contain the output probabilities of the discriminator being identified as ”false”.

However, for the networks to compete with each other, maximizing the probability of the discriminator making accurate predictions is necessary by minimizing the corresponding logarithmic probability functions. The loss function for the discriminator will then be given by:

$$lossD = -mean(log(Y_{Real})) - mean(log(1-Y_{Gen})) \tag{2}$$

where Y_{Real} will contain the output probabilities of the discriminator being identified as ”real”. That way, the score results are inversely proportional to the loss. However, it contains the same information. So, the best fit is when both scores are closets to 0.5. In this case, the discriminator could not distinguish between ”real” and ”false” images.

3.2 Conditional Generative Adversarial Network - CGAN

A CGAN is a variation of the GAN used to apply image labeling during the training process and train the generative network to generate images according to the labeling (Mirza and Osindero, 2014b). We could generate new images of a particular class in synthetic radar data, for example, only images labeled as motorcycles.

3.3 Deep Conditional Generative Adversarial Network - DCGAN

DCGAN was developed by Radford et. al (Radford et al., 2015) in 2016 and uses convolutional and convolutional-transpose layers in the discriminator and generator, respectively. In the context of automotive radar data, one of the main problems in its application in deep learning models is the lack of accessible datasets with annotations. Although labeling is one of the most challenging tasks in computer vision and its related applications, with unsupervised generative models such as GANs, a large amount of data can be generated (Abdu et al., 2021b). In this case, it is natural to think about using GAN to generate large radar signal databases and train them in an unsupervised way, helping in the labeling task.

4 EXPERIMENTAL SETUP

The identification and classification of objects is a trivial task for humans, even when the objects are hidden in the scene. However, RAD spectrum images are more complex, making it challenging to identify reflected objects. The identification is usually made with a digital camera that captures the scene simultaneously as the radar. The camera, in this case, can confirm an object class that the radar is reflecting at a given time. But, even with this support, the task is labor intensive. At the same time, to make use of radar data in classification tasks by simulators, a large set of labeled data is needed to ensure efficiency.

In this study, we want to generate new Range-Azimuth-Doppler (RAD) spectrum using GAN technique to increase the labeled image database using a human-labeled dataset. Classes were assigned and labeled using images of the scene provided by a camera system. Possible classes are labeled as pedestrian, motorcycle, truck, and car. The main objective of this article is to evaluate the ability to generate RAD images by GAN networks. Figure 2, image on the right, illustrates real RAD images, that is, constructed by the radar with the four labeled categories (car, motorcycle, pedestrian and truck). These images are the cut targets of spectrums as in Figure 1. Then, in order to identify problems and scores on how well the discriminator was able to separate new or false images, a probability statistic was used.

4.1 Accumulation of Radar Data

This study uses a short-range automotive radar operating in a frequency of 76 GHz, with coverage of up to 30 m — the radar was placed on top of a tripod, and the data was recorded in different urban scenarios. The camera was positioned next to the radar to provide image support for labeling. Figure 2 shows the positioning of the radar and camera in front of the street.

4.2 Dataset

The RAD images were labeled by a group of people using video as a visual aid. Car, motorcycle, pedestrian, and truck objects were tagged with this method, as illustrated in Figure 2. The training data is obtained by cutting out a window of 64x64 pixels around each possible object. The number of images labeled was 2680 cars, 2680 motorcycles, 2466 pedestrians and 2475 trucks.

4.3 GAN Architecture

The following generative adversarial networks were used, GAN, CGAN, and DCGAN ((Goodfellow et al., 2014), (Mirza and Osindero, 2014a), (Radford et al., 2015)), respectively). By definition of the GAN, there are two networks training together, generator and discriminator. The generator generates images from random vectors and discriminator classifies that images generated as real or fakes images (for more detail about networks architectures see (MathWorks, 2022)).

By definition of the CGAN, both the generator and discriminator have a two-input network, labels, and noise to the generator and images and labels to the discriminator that again it tries to classify the generated images as real or fake images but now based on a predefined class (see (MathWorks, 2022) for details).

The generator and discriminator defined for DCGAN also can be found in (MathWorks, 2022).

5 EXPERIMENTAL RESULTS

Extensive experiments were performed to analyze and compare the GAN architectures. The experimental analyses are divided into four phases: evaluation of the best performance between traditional GAN, CGAN, and DCGAN in generating RAD images. Assessment of the generator’s ability to produce images that can make it difficult for the discriminator to distinguish between ”real” and ”fake” images. Analysis

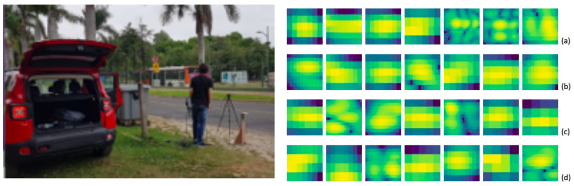

Figure 2: Positioning of the radar and camera and, example of real RAD image: a) car, b) motorcycle, c) pedestrian and d) truck.

of the influence on the values of the scores obtained by the GAN when specific parameters are changed. Furthermore, finally, analyze the applicability of the generated data for data augmentation.

In developing the GAN, CGAN, and DCGAN network, we used the architectures described earlier. The network was built using the Deep Learning Toolbox Matlab. All networks were performed with the following parameters: 100 trained epochs, 0.5 dropout, 10^{-2} learning rate, 0.5 gradient decay factor and 0.999 squared gradient decay factor. All experiments were performed on a MacBook Air 1.6 GHz Intel Core i5 Dual-Core, Intel HD Graphics 6000 1536 MB, and 8 GB 1600 MHz DDR3 memory.

To define which results obtained by the values of scores will generate the best images, the generator and discriminator score values were combined into a single value. For that, the metric used was the $L-\infty$ to determine how close both networks are in an ideal scenario. In this case, both networks should generate values close to 0.5. High values (close to 1) or shallow values (close to zero) indicate a better performance of one of the networks, invalidating the synthetic generation of images. Then, the score combined is defined by (MathWorks, 2022), where *sD* and *sG* are respectfully the *lossD* and *lossG*:

$$sC = 1 - 2 \times max(abs(sD - 0.5), abs(sG - 0.5)) \quad (3)$$

5.1 Scores Evaluation

The Table 2 presents the results obtained considering generator, discriminator and combined scores for 100 Epochs and different flipFactor. The flipFactor parameter is used to define the fraction of real labels to be inverted in the training of the discriminator. This way, it adds to the real data noise to improve the learning of the discriminator and the generator. In this case, it does not allow the discriminator to learn too quickly (MathWorks, 2022).

In our experiments, the best-combined score for the GAN network was 0.6518, with a good value for the discriminator and the generator scores. The discriminator had 0.5312 as a score, and the generator had a 0.3259 score. We can see from the graph in Figure 3 that there is a good performance between the generator and the discriminator scores. There are no peaks that indicate any significant advantage for either network. However, the lower generator score shows that some new images do not convince the discriminator properly. So, the discriminator could classify correctly between ”real” and ”fake” images, and the generator could not produce images well enough. The generated images illustrated in Figure 3 visually show the similarity with the real images (Figure 2 above). Now, with the main of improving the results of our experiments, we will add some modifications that will we discuss next.

The best-combined score for the CGAN network was 0.6391, with a high value for the discriminator score. The discriminator had a 0.6401 score, and the generator had a 0.3195 score. As we describe early, when the value of one network is close to 1.0, that indicates a better performance over another network. We can conclude, until now, that even with the good results observed, if the images generated will not necessarily be provided by a particular class of object, it is better to choose the GAN network that can cause more real data. Figure 4 illustrates the evolution graph of the generator and discriminator scores for the best score of the CGAN network, and, generated images of the motorcycle class.

Finally, the best-combined score for the DCGAN network was 0.5980, with a high value for the discriminator score again. The discriminator had a 0.6088 score, and the generator had a 0.2990 score. In this case, with a horst value of generator score in front of both GAN and CGAN generator scores. Figure 5 illustrates the evolution graph of the generator and discriminator scores for the best score of the DCGAN network, and the images generated.

The evaluation of all the images generated indicates that the best result was 0.30 flipFactor for the GAN network, and the second best result was the same flipFactor valor, 0.30, for the CGAN network. In those cases, at the same time that the discriminator learned a strong representation of characteristics that differentiate the real images from the generated images, the generator learned to represent in a very simi-

Table 2: Result of GAN and CGAN, 100 to 200 Epochs.

Network	Epoch	flipFactor	scoreD	scoreG	scoreC
GAN	100	0.30	**0.5312**	**0.3259**	**0.6518**
CGAN	100	0.30	**0.6401**	**0.3195**	**0.6391**
GAN	100	0.10	**0.5539**	**0.3069**	**0.6138**
DCGAN	100	0.30	**0.6088**	**0.2990**	**0.5980**
CGAN	100	0.50	0.4999	0.2691	0.5382
DCGAN	100	0.10	0.6869	0.2283	0.4566
DCGAN	100	0.50	0.5365	0.2061	0.4122
CGAN	100	0.10	0.7969	0.2466	0.4062
GAN	100	0.50	0.5002	0.1822	0.3644

Table 3: Scores of CGAN and GAN, with Dropout and FlipFactor Parameters.

Gan	Drop	flipF	scoreG	scoreD	scoreC	Gan	Drop	flipF	scoreG	scoreD	scoreC
CGAN	0.75	0.1000	**0.4904**	**0.5636**	**0.8728**	GAN	0.75	0.3000	**0.3535**	**0.5495**	**0.7070**
CGAN	0.50	0.5000	**0.2633**	**0.5278**	**0.5266**	GAN	0.75	0.1000	**0.3365**	**0.5937**	**0.6731**
CGAN	0.75	0.3000	0.2618	0.5359	0.5236	GAN	0.50	0.3000	0.3195	0.6401	0.6391
CGAN	0.50	0.3000	0.2446	0.6342	0.4892	GAN	0.25	0.1000	0.2565	0.5921	0.5130
CGAN	0.75	0.5000	0.2305	0.5033	0.4610	GAN	0.75	0.5000	0.2257	0.5251	0.4515
CGAN	0.50	0.1000	0.2115	0.6604	0.4230	GAN	0.50	0.5000	0.2121	0.5527	0.4243
CGAN	0.25	0.1000	0.1963	0.8500	0.3000	GAN	0.50	0.1000	0.1922	0.7041	0.3844
CGAN	0.25	0.3000	0.1306	0.7412	0.2612	GAN	0.25	0.5000	0.1805	0.6265	0.3611
CGAN	0.25	0.5000	0.0760	0.5535	0.1520	GAN	0.25	0.3000	0.1478	0.6859	0.2956

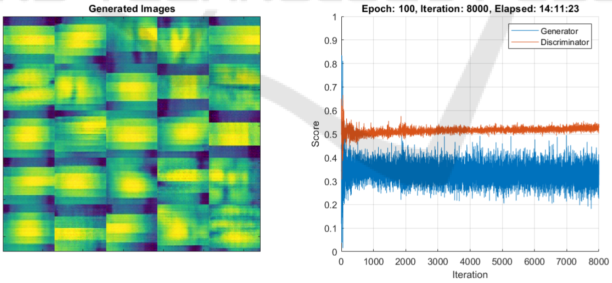

Figure 3: Generated images with GAN, 0.30 flipFactor; Graphic of scores results from evolution for 100 Epochs.

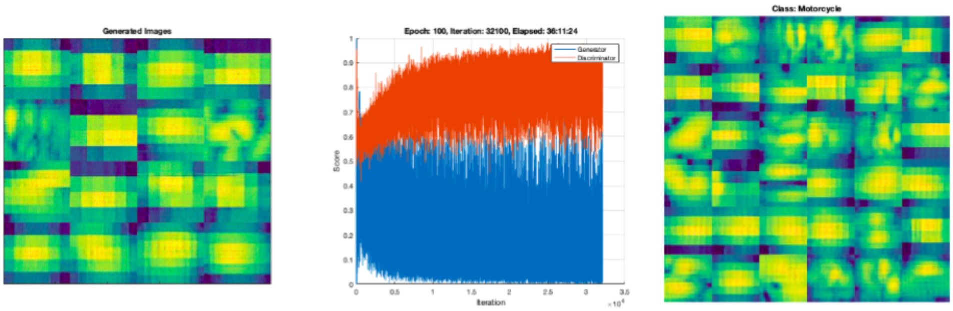

Figure 4: Generated images with CGAN, 100 epochs, 0.30 flipFactor, graphic of scores results and motorcycle class.

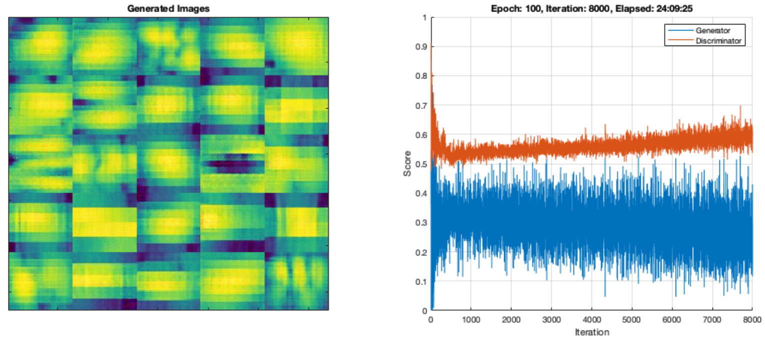

Figure 5: Generated images with DCGAN, graphic of scores results.

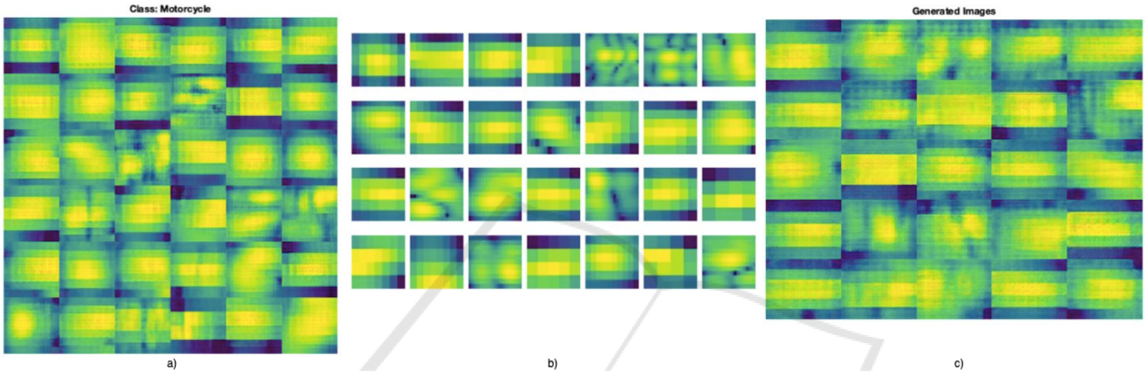

Figure 6: The best image generated: a) CGAN (0.8728 score, only motorcycle class sample), b) Real images and c) GAN (0.7070 score, all classes sample).

lar way the characteristics that allow generating images to be similar to the training data. It is possible to observe that the discriminator is improving the classification while the generator is improving the ability to generate new convincing images. However, the question remains whether it is possible to use those generated images to augment new data-set.

The final set of experiments included changes in dropout parameters (Hinton et al., 2012). Our previous work at Skeika, E. et al. (Skeika et al., 2020) describes how the dropout technique is essential in reducing over-fitting during training. In this sense, the networks were trained with the following dropout values, 0.25, 0.50, and 0.75, and combined with different values of flipFactor and 50 Epochs. We decided to change the number of Epochs because, in this part of our experiments, the main objective is to analyze the influence of changes values of those parameters on the final generator and discriminator scores. With those experiments, we intend to answer how changes in dropout and flipFactor values can modify the scores of the two better networks previously observed, GAN and CGAN. Furthermore, determine the applicability of the generated data for data augmentation.

According to Table 3, we can observe an evolution in the network performance after each modification proposed in its architecture. We choose only GAN and CGAN for testing because of their better scores in previous analyses than those values by DCGAN. We can highlight three of them: for dropout values of 0.75 and flipFactor of 0.10, the combined score value is 0.8728 by CGAN; for dropout 0.75 and flipFactor 0.30, the combined score value is 0.7070, but now, by GAN; and, dropout of 0.75 and flipFactor of 0.10 resulted in a combined score of 0.6731 by GAN again. Next, the best score combined is again by GAN, 0.6391.

Despite values 0.8728, 0.7070, 0.6731, and 0.6391 of the combined scores being high, these results are not enough to attest to the quality of the new images generated and the ability of the discriminator in the classification. When we observe the scores individually, we have clues about the behavior of the networks and, consequently, the performance in the competition between them. The generator score in these cases ideally can not be much lower than the discriminator score. Despite the generator generating good images, when the value score of the discriminator is much higher, it can indicate that it still correctly classifies most images.

So, if we look at scores of the first best-combined score, that is, 0.8728, we can infer that there is a better response from both the generator and the discriminator. Both values are balanced (0.4904 - generator; 0.5636 - discriminator), which suggests a slight advantage in the discriminator classification. However, with the excellent generator score (98% of the ideal score), the discriminator will classify many ”fake” images as ”real,” which is our ultimate goal.

The future works

Regarding the modified parameters, dropout, and flipFactor, we can observe the impact of these changes on the final performance of both networks. In our experiments, the best results were obtained using a value of 0.75 for dropout for the three better scores, followed by a 0.50 value for the higher scores observed. Nevertheless, for flipFactor values, it was impossible to decide what value fits better. Unfortunately, for flipFactor parameter value, the only way to decide on the better value is to test all possibilities for different contexts. We provide a set of experiments showing how dropout and flipFactor could influence the final scores on generator and discriminator training. The dropout played more influence on generator performance, so it is necessary to include high dropout values to avoid overfitting and generate more realistic images. However, the flipFactor, just empirical experiments, can decide what parameter value is better to influence on final scores.

One final observation we can include is the scores obtained with the same parameters in both GAN and CGAN networks but with different numbers of Epochs. Observe in Table 1 that used 0.5 dropouts and 0.3 flipFactor. We said that the best network was CGAN based on its score value. We can observe that in Table 1 and Table 2, in which Epochs decrease to 50, the scores values change but remain whit the better score by GAN when we look at the 0.5 dropouts and 0.3 flipFactor value. On the other hand, as we said, the dropout parameters can modify the final score for both GAN and CGAN, shown the better final score of CGAN. The Figure 6 shows images generated by the best scores obtained by CGAN (0.8728, as the choice for the model, images of the motorcycle class were generated) and by GAN (0.7070, with an example of all classes mixed).

6 CONCLUSIONS

In this paper, we experimentally evaluated the training of GAN for synthetic RAD image generation. This evaluation adds a new possibility for data augmentation when data labeling available is not enough. The results showed that, yes, the GAN generated RAD images well, even when a specific class of the object is specified. We also compared the scores of three GAN architectures, GAN, CGAN, and DCGAN, in RAD synthetic imaging generation. CGAN scored the highest, while GAN scored the second highest score but the better scores in most experiments. DCGAN performed lower, so we cut it to the final experiments. We show that the generator can produce RAD images well enough with the results analyzed. Consequently, it making it difficult for the discriminator to distinguish between ”real” and ”fake” images. We can use the RAD images generated to improve the simulator experiments, including a large dataset with those images, adding data augmentation. This work also opens opportunities for new studies related to RAD data classification, such as conditional generations based on vehicle class.

In future work, we intend to analyze whether or not a machine learning classifier will classify RAD images better by adding GAN RAD synthetic image as data augmentation create possibilities to the development of . We also would like to evaluate the synthetic image generation by GAN for radar micro doppler.

ACKNOWLEDGMENTS

This study was supported by the Fundação de Desenvolvimento da Pesquisa - Fundep Rota 2030 (27192*09).

REFERENCES

Abdu, F. J., Zhang, Y., Fu, M., Li, Y., and Deng, Z. (2021a). Application of deep learning on millimeter-wave radar signals: A review. *Sensors*, 21(6).

Abdu, F. J., Zhang, Y., Fu, M., Li, Y., and Deng, Z. (2021b). Application of deep learning on millimeter-wave radar signals: A review. *Sensors*, 21(6):1951.

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., and Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938.

Alnujaim, I., Ram, S. S., Oh, D., and Kim, Y. (2021). Synthesis of micro-doppler signatures of human activities from different aspect angles using generative adversarial networks. *IEEE Access*, 9:46422–46429.

Danielsson, L. (2010). *Tracking and radar sensor modelling for automotive safety systems*, volume Doctoral d.

de Oliveira, M. L. L. and Bekooij, M. J. G. (2020). Generating synthetic short-range fmcw range-doppler maps using generative adversarial networks and deep convolutional autoencoders. In *2020 IEEE Radar Conference (RadarConf20)*, pages 1–6.

Deep, Y. (2020). Radar cross-sections of pedestrians at automotive radar frequencies using ray tracing and point scatterer modelling. *IET Radar, Sonar & Navigation*, 14:833–844(11).

Doherty, H. G., Cifola, L., Harmanny, R. I. A., and Fioranelli, F. (2019). Unsupervised learning using generative adversarial networks on micro-doppler spectrograms. In *2019 16th European Radar Conference (EuRAD)*, pages 197–200.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017). CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16.

Eder, T., Hachicha, R., Sellami, H., van Driesten, C., and Biebl, E. (2019). Data driven radar detection models: A comparison of artificial neural networks and non parametric density estimators on synthetically generated radar data. In *2019 Kleinheubach Conference*, pages 1–4.

Erol, B., Gurbuz, S. Z., and Amin, M. G. (2019). Gan-based synthetic radar micro-doppler augmentations for improved human activity recognition. In *2019 IEEE Radar Conference (RadarConf)*, pages 1–5.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors.

Iberle, J., Mutschler, M. A., Scharf, P. A., and Walter, T. (2019). A radar target simulator concept for close-range targets with micro-doppler signatures. In *2019 12th German Microwave Conference (GeMiC)*, pages 198–201.

Javadi, S. H. and Farina, A. (2020). Radar networks: A review of features and challenges. *Information Fusion*, 61:48–55.

Jiao, L. and Zhao, J. (2019). A survey on the new generation of deep learning in image processing. *IEEE Access*, 7:172231–172263.

Kern, N. and Waldschmidt, C. (2022). Data augmentation in time and doppler frequency domain for radar-based gesture recognition. In *2021 18th European Radar Conference (EuRAD)*, pages 33–36.

Lee, H., Ra, M., and Kim, W.-Y. (2020). Nighttime data augmentation using gan for improving blind-spot detection. *IEEE Access*, 8:48049–48059.

Lekic, V. and Babic, Z. (2019). Automotive radar and camera fusion using generative adversarial networks. *Computer Vision and Image Understanding*, 184:1–8.

Liu, G., Wang, L., and Zou, S. (2017). A radar-based blind spot detection and warning system for driver assistance. In *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pages 2204–2208.

Magosi, Z. F., Li, H., Rosenberger, P., Wan, L., and Eichberger, A. (2022). A survey on modelling of automotive radar sensors for virtual test and validation of automated driving. *Sensors*, 22(15).

MathWorks (2022). Use experiment manager to train generative adversarial networks (gans). url-https://www.mathworks.com/help/deeplearning.

Mirza, M. and Osindero, S. (2014a). Conditional generative adversarial nets.

Mirza, M. and Osindero, S. (2014b). Conditional generative adversarial nets.

Ngo, A., Bauer, M. P., and Resch, M. (2021). A multi-layered approach for measuring the simulation-to-reality gap of radar perception for autonomous driving.

Qu, L., Wang, Y., Yang, T., Zhang, L., and Sun, Y. (2021). Wgan-gp-based synthetic radar spectrogram augmentation in human activity recognition. In *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pages 2532–2535.

Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks.

Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks.

Rahman, H. (2019). *Fundamental Principles of Radar*. Cambridge: Cambridge University Press.

Rahman, M. M., Gurbuz, S. Z., and Amin, M. G. (2021). Physics-aware design of multi-branch gan for human rf micro-doppler signature synthesis. In *2021 IEEE Radar Conference (RadarConf21)*, pages 1–6.

Rahman, M. M., Malaia, E. A., Gurbuz, A. C., Griffin, D. J., Crawford, C., and Gurbuz, S. Z. (2022). Effect of kinematics and fluency in adversarial synthetic data generation for ASL recognition with RF sensors. *IEEE Transactions on Aerospace and Electronic Systems*, 58(4):2732–2745.

Schuler, K., Becker, D., and Wiesbeck, W. (2008). Extraction of virtual scattering centers of vehicles by ray-tracing simulations. *IEEE Transactions on Antennas and Propagation*, 56(11):3543–3551.

Skeika, E. L., Luz, M. R. D., Fernandes, B. J. T., Siqueira, H. V., and De Andrade, M. L. S. C. (2020). Convolutional neural network to detect and measure fetal skull circumference in ultrasound imaging. *IEEE Access*, 8:191519–191529.

Song, Y., Wang, Y., and Li, Y. (2019). Radar data simulation using deep generative networks. *The Journal of Engineering*, 2019(20):6699–6702.

Vishwakarma, S., Tang, C., Li, W., Woodbridge, K., Adve, R., and Chetty, K. (2021). Gan based noise generation to aid activity recognition when augmenting measured wifi radar data with simulations. In *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6.

Wald, S. O. and Weinmann, F. (2019). Ray tracing for range-doppler simulation of 77 ghz automotive scenarios. In *2019 13th European Conference on Antennas and Propagation (EuCAP)*, pages 1–4.

Wheeler, T. A., Holder, M., Winner, H., and Kochenderfer, M. (2017). Deep stochastic radar models.

Yun, Z. and Iskander, M. F. (2015). Ray tracing for radio propagation modeling: Principles and applications. *IEEE Access*, 3:1089–1100.