IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium | 978-1-7281-6374-1/20/$31.00 ©2020 IEEE | DOI: 10.1109/IGARSS39084.2020.9323676

RADAR SENSOR SIMULATION WITH GENERATIVE ADVERSARIAL NETWORK

Maryam Rahnemoonfar [1], Masoud Yari [1], John Paden [2]

1. Computer Vision and Remote Sensing Laboratory, University of Maryland, Baltimore County, USA
2. Center for Remote Sensing of Ice Sheets, University of Kansas, Kansas, USA.

ABSTRACT

Significant resources have been spent in collecting and storing large and heterogeneous radar datasets during expensive Arctic and Antarctic fieldwork. The vast majority of data available is unlabeled, and the labeling process is both time-consuming and expensive. One possible alternative to the labeling process is the use of synthetically generated data with artificial intelligence. In this research, we evaluated the performance of synthetically generated snow radar images based on modified cycle-consistent adversarial networks. We conducted several experiments to test the quality of the generated radar imagery. Our experiments show a very good similarity between real and synthetic snow radar images.

Index Terms—
convolutional neural network, generative adversarial network, ice tracking, radar imagery

1. INTRODUCTION

The melting of polar ice sheets makes a considerable contribution to ongoing sea-level rise and changing ocean circulation, leading to coastal flooding and risking the homes and livelihoods of tens of millions globally. Recent large-scale radar surveys of Greenland and Antarctica reveal internal ice layers on a continental scale enabling accurate detection and tracing of these internal layers to illuminate many aspects of ice sheet dynamics, including their history and their response to climate and subglacial forcing. In this study we have used snow radar data produced by the Center for Remote Sensing of Ice Sheets for NASA Operation IceBridge. The snow radar is a profiling instrument which produces vertical sounding images of snow layers over ice sheets and ice caps. The radar signal is sensitive to annual density changes that occur due to the seasonal transitions from summer to winter; this density change interface scatters the radar signal which is measured by the radar's receiver.

Several semi-automated and automated methods exist for surface and bottom tracking in radar images [1, 2, 3, 4, 5, 6, 7]. Tracking internal layers is a significantly more difficult task because of the large number of layers in close proximity. However the traditional techniques which are semi-supervised and based on feature engineering cannot be scaled up to big data.

In recent years, deep learning tasks have been developed for several tasks such as classification [9], object recognition [10], counting [11, 12], and semantic segmentation [13]. However, these algorithms are limited to cases where large labeled datasets are available.

The vast majority of data available in the remote sensing community is unlabeled, and the labeling process is both time-consuming and expensive. One possible alternative to the labeling process is the use of synthetically generated data.

The standard way for generating synthetic radar data (such as from a radar depth sounder used to measure ice thickness) is to simulate the radar scattering response using digital elevation models (DEM) of the ice surface and bottom. Usually, the DEM is represented by a sheet of points or facets and the total scattering response is the superposition of the scattering from all of these targets. This data can then be processed through the regular radar data processing chain to produce a simulated radar image. However, data generation based on a physics simulation is compute-intensive and cannot be used for generating large data sets.

The goal of this research is to generate synthetic snow radar images that can be used to train data-driven algorithms such as deep convolutional neural nets. In this research, we developed a data-driven machine learning approach, Generative Adversarial Networks (GANs), for generating synthetic radar data. A GAN [14] is composed of two simultaneously trained parts called a generator and discriminator. The discriminator is trained to tell the difference between real and fake images. The generator is trained to generate realistic-looking images and fool the discriminator. Both components improve until the synthetic images are indistinguishable from the real images. The discriminators accuracy reduces to 0.5, indicating that is simply guessing when it makes its decision.

For generating synthetic radar images from labeled data we used a conditional GAN [15]. Figure 1-right shows the synthetic snow radar image which is generated from labeled data. Contours show the internal layers of ice sheet. (Figure 1-left),

The CycleGAN network works with two different sets of images. Each set of images has its own discriminator. Two different mapping functions are used, called G and F. Each mapping function translates an image from one set to

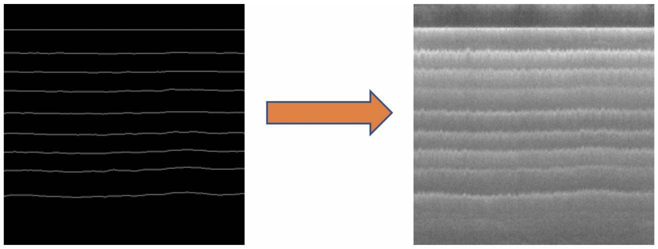

Fig. 1. Generated snow radar image (**right**) from the label data representing internal layers of ice sheet (**left**). The horizontal axis is the flight path and the vertical axis is the ice depth.

the other. The network works with the intuition that if an image from one set is translated to the other, and that the resulting image is translated back to the original set, the final result should be approximately the same as the original. The difference in these images is termed cycle-consistency loss, and is what the network tries to minimize.

2. METHODOLOGY

The initial goal inherited from the CycleGAN is to learn the mapping functions between the two domains X and Y [15]. In the first stage, adversarial loss [14] is applied to asses each generator that will be responsible for learning one domain: $G : X \to Y$ and $F : Y \leftarrow X$. Additionally each mapping function will also be paired with a discriminator D_X and D_Y. The adversarial loss [14] for the mapping function can expressed as [15]:

$$\begin{aligned}\mathcal{L}_{GAN}(G, D_Y, X, Y) = E_{y\sim pdata}(y)[logD_Y(y)] \\ +E_{x\sim pdata}(x)[log(1 - D_Y(G(x)))],\end{aligned} \tag{1}$$

where G attempts to generate images $G(x)$ that look similar to the images of domain Y, while D_Y will try to discriminate between the translated samples $G(x)$ and the real samples y. G's goal is to minimize this objective against its adversary D while D will aim to maximize it. Using this we can also apply it to another mapping function $F : Y \to X$ and its discriminator D_X[15]. Using the adversarial loss these networks can, in theory, learn mappings G and F that produce outputs identical to the target domains X and Y if G and F are stochastic functions [16]. However, with large enough capacity, a network can map the same set of input images to any random permutation of images in the target domain, where any of the learned mappings can induce an output distribution that matches the target distribution. The cycle constancy loss is defined according to the following formula:

$$\begin{aligned}\mathcal{L}_{cyc}(G, F) = E_{x\sim pdata}(x)[\|F(G(x)) - x\|] \\ +E_{y\sim pdata}(y)[\|G(F(y)) - y\|].\end{aligned} \tag{2}$$

The full objective that is inherited from CycleGAN [15] is the combination of the two losses with an addition of the λ parameter that control importance of the losses.

$$\begin{aligned}\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{GAN}(G, D_Y, X, Y) \\ +\mathcal{L}_{GAN}(F, D_X, Y, X) + \lambda\mathcal{L}_{cyc}(G, F).\end{aligned} \tag{3}$$

Each of the GAN losses apply mean squared error as the criterion. Training the G to minimize $E_{x\sim pdata}(x)[D(G(x)) - 1]^2$ and train the D to minimize $E_{y\sim pdata}(y)[D(y) - 1]^2 + E_{y\sim pdata}(x)[D(G(x))]^2$ [15]. This proves to be suitable for images that have larger values or have less discrete values. Additionally, it is also a suitable criterion for generalization.

For the discriminator we maintain the network used by CycleGAN [15] which is a convolutional neural network with five convolutions. Using these residual connections shows higher stability in unpaired image to image translation, versus a U-net type network with the skip connections based on an encoding and decoding pathway. This can lead to instability with unpaired training due to less information being available.

3. EXPERIMENTAL RESULTS

The images used in this research are CReSIS standard output products collected with snow radar in 2012. The horizontal axis is along the flight path and the vertical axis represents depth. To train the CycleGAN network we used the snow radar dataset and the ground-truth images which are produced by human annotators.

Our snow radar dataset includes images of various sizes. Our dataset is comprised of 2,361 training images and 260testing images and the annotated data associated with them.

3.1. Qualitative Results

The result of generating synthetic images from ground-truth labels are displayed in Figure 2. In this figure, the inputs to the algorithm are labels (left column). The Generator generated the synthetic images using the input labels (middle column). The generated synthetic images look very much like real radar images (right column). Even though all annotated labels have the same intensity, still generated images can highlight some layers more than others exactly similar to real radar data. For example, in Figure 2, the third row shows that the middle layers are brighter than other layers, even though their labels are the same brightness (Figure 2-Left). We see the similar patterns in the real image.

When the labeled data contains all of the internal layers, GAN can create perfect synthetic data. However, our labeled data is not perfect and there are some missing layers or some incomplete layers (Figure 3). In this case, our model can generate some arbitrary lines but usually they do not follow the same pattern as real images.

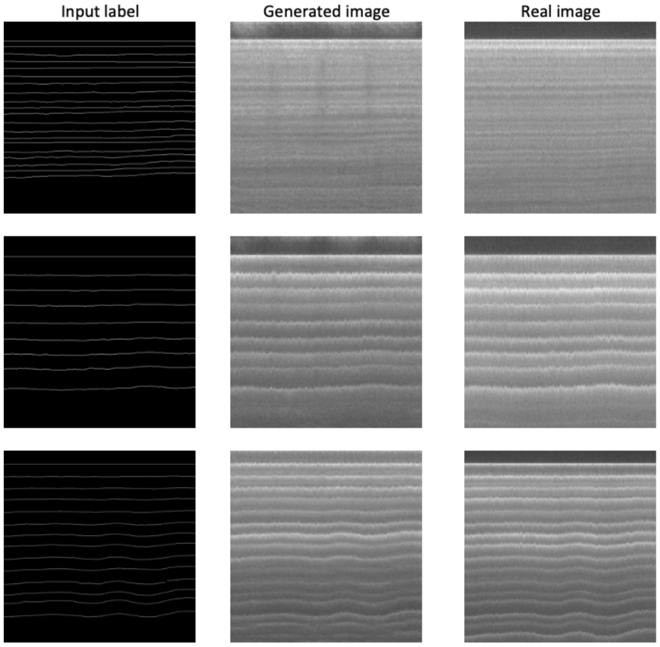

Fig. 2. Label to Image results paired with the actual radar image (**right**) for comparison. The input to the network is the label (**left**) while it generates the image (**middle**).

3.2. Quantitative Results

To evaluate our results quantitatively, we include two metrics namely structural similarity index (SSIM) [17] and peak signal to noise ratio (PSNR).

The structural similarity index is expressed as [17]:

$$SSIM(x, y) = I(x, y)^{\propto} C(x, y)^{\beta} S(x, y)^{\gamma}, \tag{4}$$

where I is luminance, C is contrast, and S is structure [17]. The SSIM attempts to model the structural change of an image by comparing small windows or sub-samples in the image to compare the luminance, contrast, and structure of the two images [17]. This metric gives us a robust measure of the perceived changes in the image. The closer the SSIM is to 1.0 the higher the quality image we have [17]. Another evaluation metric is peak signal to noise ratio (PSNR) or signal to noise ratio (SNR) which is commonly used in the signal processing area as an image quality metric. PSNR is expressed as [18]:

$$PSNR = 10 \log_{10} \frac{255^2}{\langle n(x, y)^2 \rangle}, \tag{5}$$

where $\langle n(x, y)^2 \rangle$ gives mean square error [18]. The higher the PSNR (in dB), the better the quality of the generated image. Using these two metrics we can show how well the generated synthetic images look like the real images.

SSIM and PSNR values are depicted in Table 1. The maximum SSIM value is close to 1 which means two sets of images are very similar for some images. However the average value is quite low due to missing layers in our labeled dataset. PSNR is usually around 20 dB for the visible images generated by GAN algorithm. PSNR for Radar imagery is usually between 25–30dB. This shows that our generated images have similar noise content and comparable image quality to real radar imagery.

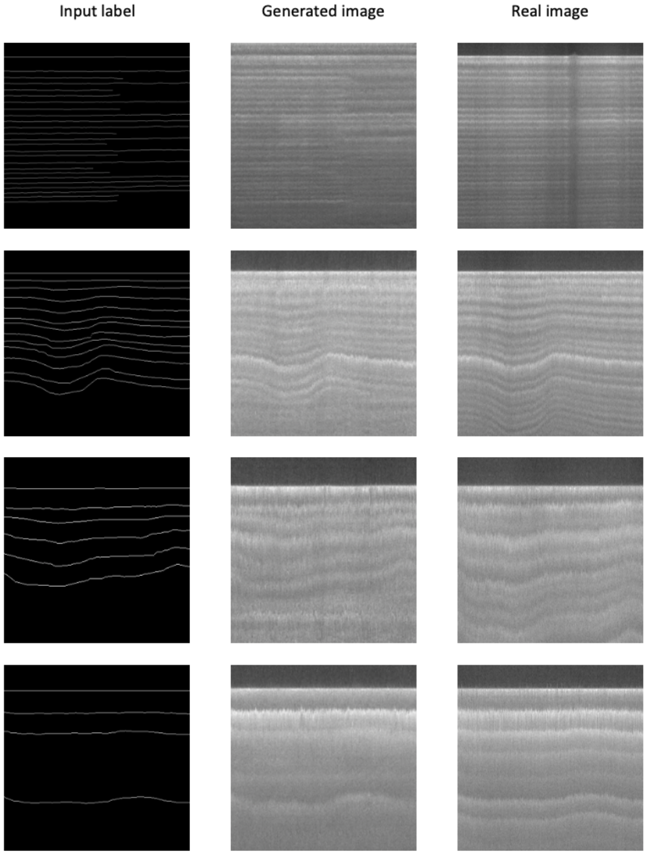

Fig. 3. Generating images with missing labels- Label to Image results paired with the actual radar image (**right**) for comparison. The input to the network is the label (**left**) while it generates the image (**middle**).

Table 1. Evaluation results

	Min	Average	Max
SSIM	0.363	0.569	0.721
PSNR	11.22	21.18	28.67

4. CONCLUSION

Here we developed an architecture based on the CycleGAN network to generate synthetic snow radar images. This method can also be used to generate other types of radar images. We conducted several experiments for testing the generated synthetic images based on qualitative similarity metrics. Similarity metrics demonstrate a good statistical proximity of AI-generated results to the real radar data. Moving forward we explore the combination of AI and physics simulators for a more realistic radar data simulator.

5. ACKNOWLEDGMENTS

This work is partially supported by NSF BIGDATA awards (IIS-1838230, IIS-1838024), IBM, and Amazon.

6. REFERENCES

[1] D. J. Crandall, G. C. Fox, and J. D. Paden, "Layer-finding in radar echograms using probabilistic graphical models," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 1530–1533.

[2] S. Lee, J. Mitchell, D. J. Crandall, and G. C. Fox, "Estimating bedrock and surface layer boundaries and confidence intervals in ice sheet radar imagery using mcmc," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 111–115.

[3] J. E. Mitchell, D. J. Crandall, G. C. Fox, M. Rahnemoonfar, and J. D. Paden, "A semi-automatic approach for estimating bedrock and surface layers from multichannel coherent radar depth sounder imagery," in *SPIE Remote Sensing*. International Society for Optics and Photonics, Conference Proceedings, pp. 88 921–88 926.

[4] J. E. Mitchell, D. J. Crandall, G. Fox, and J. Paden, "A semi-automatic approach for estimating near surface internal layers from snow radar imagery," in *IGARSS*, Conference Proceedings, pp. 4110–4113.

[5] M. Rahnemoonfar, M. Yari, and G. C. Fox, "Automatic polar ice thickness estimation from sar imagery," in *SPIE Defense+ Security*. International Society for Optics and Photonics, 2016, pp. 982 902–982 902.

[6] M. Rahnemoonfar, G. C. Fox, M. Yari, and J. Paden, "Automatic ice surface and bottom boundaries estimation in radar imagery based on level-set approach," *IEEE Transactions on Geoscience and Remote Sensing*, 2017.

[7] M. Rahnemoonfar, A. Abbassi, J. Paden, and G. C. Fox, "Automatic ice thickness estimation in radar imagery based on charged particle concept," *IEEE International Geoscience and Remote Sensing Symposium*, 2017.

[8] C. Panton, "Automated mapping of local layer slope and tracing of internal layers in radio echograms," *Annals of Glaciology*, vol. 55, no. 67, pp. 71–77, 2014.

[9] C. Sheppard and M. Rahnemoonfar, "Real-time scene understanding for uav imagery based on deep convolutional neural networks," in *Geoscience and Remote Sensing Symposium (IGARSS), 2017 IEEE International*. IEEE, 2017, pp. 2243–2246.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 580–587.

[11] M. Rahnemoonfar and C. Sheppard, "Deep count: fruit counting based on deep simulated learning," *Sensors*, vol. 17, no. 4, p. 905, 2017.

[12] M. Rahnemoonfar, D. Dobbs, M. Yari, and M. J. Starek, "Discountnet: Discriminating and counting network for real-time counting and localization of sparse objects in high-resolution uav imagery," *Remote Sensing*, vol. 11, no. 9, p. 1128, 2019.

[13] M. Rahnemoonfar, M. Robin, M. V. Miguel, D. Dobbs, and A. Adams, "Flooded area detection from uav images based on densely connected recurrent neural networks," in *Geoscience and Remote Sensing Symposium (IGARSS), 2017 IEEE International*. IEEE, 2018, pp. 3743–3746.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[15] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2017.244

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.

[17] K. Ridgeway, J. Snell, B. Roads, R. S. Zemel, and M. C. Mozer, "Learning to generate images with perceptual similarity metrics," *CoRR*, vol. abs/1511.06409, 2015. [Online]. Available: http://arxiv.org/abs/1511.06409

[18] S. Yao, W. Lin, E. Ong, and Z. Lu, "Contrast signal-to-noise ratio for image quality assessment," in *IEEE International Conference on Image Processing 2005*, vol. 1, Sep. 2005, pp. I–397.