Range-Doppler Map Augmentation by Generative Adversarial Network for Deep UAV Classification

SeongUk Park
Convergence Science and Technology
Seoul National University
Seoul, South Korea
swpark0703@snu.ac.kr

Seungeui Lee
Convergence Science and Technology
Seoul National University
Seoul, South Korea
seungeui.lee@snu.ac.kr

Nojun Kwak
Convergence Science and Technology
Seoul National University
Seoul, South Korea
nojunk@snu.ac.kr

***Abstract*—In this paper, we report a data augmentation technique to alleviate the data shortage for UAV (Unmanned Aerial Vehi-cle) classification problem. The UAV classification problem is modeled on CNN (Deep Convolutional Neural Network), which is prevalent in artificial intelligence, and the training data consists of an RD (Range-Doppler) map of a FMCW (Frequency-Modulated Continuous-Wave) radar for a moving UAV. Getting more training data usually helps a deep CNN better generalize to test data or the real world. Therefore, we introduce a Generative Adversarial Network (GAN)-based data augmentation technique to generate synthetic RD maps used for training of UAV classifiers. By doing so, the UAV classifier was able to achieve better performance on the test dataset, especially when the classifier was trained on a smaller dataset.**

***Index Terms*—UAV, classification, data augmentation, synthetic data, convolutional neural network, generative adversarial network, Range-Doppler, FMCW, radar**

I. INTRODUCTION

Radar sensors are nowadays an important tool to replace the use of vision-based sensors such as cameras, especially in low light and under certain conditions such as harsh weather. Because of their robustness and versatility, radar sensors are used complementary to vision sensors such as LIDAR [1]–[3] in many tasks such as autonomous driving and unmanned aerial vehicle (UAV) detection.

Recently, artificial intelligence technology using deep learning (DL) has recorded state-of-the-art performance in various sensing and inference tasks, and is surpassing human performance in image classification [11]. The remarkable performance of DL models in classification comes from the use of convolution filters in neural networks, which are called convolutional neural networks (CNNs). CNNs are widely used in Generative Adversarial Networks (GANs) [12], [13], especially among deep generative models.

The growth of artificial intelligence has made it possible to apply DL models to radar signals for a variety of tasks as well. GAN-based generation tasks such as denoising [4] and super-resolution [5] or CNN-based UAV classification [6] are used for short-time Fourier transform (STFT) radar spectrogram signals and CNN-based moving object detection [7] is used for RD map signals and many other [8]–[10].

Unlike other tasks, the task of classifying objects using radar signals deserves special attention. Other tasks that use DL are limited to helping the human senses to observe results using the output of radar sensors, but classifying using DL models allows the model to make decisions directly without human detection. To do this, the DL model must be reliable in the real environment (test data). This means that the model must be very accurate in order to be used. However, in general, DL models require large amounts of training data to achieve high generalization performance, which is difficult to obtain.

In this paper, we propose a conditional GAN [14]-based RD map data augmentation method that improves the performance of the RD map classifier. Our GAN model is trained using the collected UAV RD maps and generates a synthetic UAV RD map according to a given input class. By using the synthetic UAV RD map as additional training data, the trained classifier can achieve higher classification performance, especially in data-poor environments. The contributions of our research are:

- We collected RD map information for the free flight of three different types of UAVs, and successfully trained a decent CNN classifier that performs R-D map classification.
- We successfully trained conditional GAN that successfully generates plausible synthetic RD map, using UAV RD map dataset.
- By using the synthetic RD map data, we boosted the classification performance of UAV RD map classifier.

The rest part of our paper is as following: In section II, we explain our training and testing data acquisition process. In section III, we provide detailed explanation for the training of CNN classifier and conditional GAN model. In the rest of the sections, we provide detailed settings of our model training and report experiment results of our trained model.

II. DATASET

In order to train our CNN classifier and GAN models, FMCW dataset Range-Doppler maps of UAVs are required.

However, since UAV radar data are generally collected for a particular purpose by specific companies or military institutions, they are not publicly available for academic use, so that researchers generally collect using their equipment. For military purposes, Range-Doppler maps are utilized to extract both the distance and radial velocity of the target. With an FMCW radar signal, a 2-dimensional FFT (fast Fourier transform) is typically used to generate a Range-Doppler map. Specifically, a Range-FFT is applied over the samples of each chirp signal and then a Doppler-FFT is applied across chirps. After 2D FFT processing, we take the magnitude of a complex Range-Doppler array. For UAVs, three types of four-wing drones of different sizes were used. For each drone, we obtained data from multiple free-flight shots with radar equipment. We also acquired radar data for people and cars, but these two data were excluded because it was too easy for the UAV classifier to distinguish them from the rest of the classes.

III. Proposed Model Training

In this paper, we propose to use a GAN model to generate a synthetic RD map of UAVs to aid in the training of UAV classifiers. Our training method consists of 3 steps: 1) cGAN (Conditional Generative Adversarial Network) training, 2) synthetic data augmentation using cGAN, and 3) UAV classifier training utilizing augmented data. This section details the three-step learning process.

A. Conditional Generative Adversarial Network Training

In the field of deep learning, Generative Adversarial Networks (GANs) have been adopted to diverse vision tasks [15]–[19]. In general, GANs are composed of two networks: a generator network G and a discriminator network D. Given a vector $z \in R^{100}$ whose value of each element are sampled from unit normal distribution as input, the G outputs an image vector that is trained to mimic the real distribution of the real data $x \in R^{1\times128\times128}$, which is fake data $G(z) \in R^{1\times128\times128}$. Given the real data x and fake data $G(z)$, D learns to discriminate the x as real and $G(z)$ as fake, and G learns to fool D to classify $G(z)$ as real. This is modeled as min-max game, which the value function is:

$$\begin{aligned} \underset{G}{min}\,\underset{D}{max}\, V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] \\ + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \end{aligned} \tag{1}$$

During the training of the two network, G and D compete each other, and this process is often compared to counterfeit money criminal and the police. The G will try to make the fake samples more real to fool the discriminator, and the discriminator will learn better strategies that can discriminate real sample and fake sample. Ideally, if the distribution of the fake samples becomes identical to the real data distribution, the probability of the fooling rate of the generator will become 0.5 and the accuracy of the discriminator will become 0.5, which is the point of equilibrium. However, such ideal case rarely holds in practice. Currently in the field of deep learning, with equal conditions, the discriminative models usually optimize faster than the generative models. As a result, the discriminator delivers strong gradients to the generator, which is harmful for the learning of the generator.

Consequently, many researchers devised ways to make the competitive process more stable, so that the generator and the discriminator can make synergy each other. Among those we adopt Least-Square GAN (LSGAN) [20] for our loss function, which proposed mean-squared-error (MSE) loss for the stable training of GAN. The two network G and D are jointly trained to minimize these two loss functions:

$$\begin{aligned} L_D &= 0.5[[D(G(z))]^2 + [1-D(x)]^2] \\ L_G &= 0.5[[1-D(G(z))]^2] \end{aligned} \tag{2}$$

In the setting of our method, we adopt the training strategy of Auxiliary Classifier GAN (ACGAN) [21], which trains the G to be capable of generating fake sample according to the conditioned target class label $c \in R^3$. The target label c is used as the input condition of the network G with the noise vector z given as input, being $G(z|c)$. For the discriminator, its task-head is modified to conduct multi-task learning: 1) discrimination task-head, which discriminates real data x and fake data $G(z|c)$, and 2) classification task-head, which learns to predict the correct label of x or $G(z|c)$. The G learns to generate $G(z|c)$ that fools the D to classify it to correct label y. We name the classification head of the discriminator as D_C. By using softmax-cross-entropy loss function, the additional loss function becomes:

$$\begin{aligned} L_{D_C} &= -\log p(c|x) \\ L_{G_C} &= -\log p(c|G(z|c)) \end{aligned} \tag{3}$$

Combining Eq. (2) and Eq. (3) together, the total loss L_{total} for the training of G and D is:

$$L_{total} = L_D + L_{D_C} + L_G + L_{G_C}, \tag{4}$$

which ends up with combination of conditional GAN and LSGAN. In the default settings above, we added two more details to stabilize the training of the GAN.

First is the conditioning algorithm of the class label c. Assignment of conditions to GAN is first proposed by cGAN [14] to use class labels as one-hot vectors by concatenation operation with inputs of G and D, which becomes $G(z;y)$ and $D(G(z);y)$. Due to its simplicity, the use of concatenation operations has been shown in many other GAN studies such as ACGAN [21] and StarGAN [22]. In our proposed method, we adopt the CCAM embedding operation of SGN [23]. CCAM uses additional multi-layer-perceptron (MLP) layers which use intermediate feature representation $f \in R^{1\times1\times ChannelSize}$ which is spatially average-pooled version of $F \in R^{H\times W\times ChannelSize}$and class one-hot vector as input, and its output is activated by sigmoid operation S, which formulates:

$$\begin{aligned} &CCAM(f,c) = S(MLP_3\,(MLP_2\,(f; MLP_1(c)))) \\ &F \leftarrow F \otimes CCAM(f,c) \end{aligned} \tag{5}$$

Where $\otimes$ denotes element-wise multiplication operation in the channel dimension. We found that this conditioning method

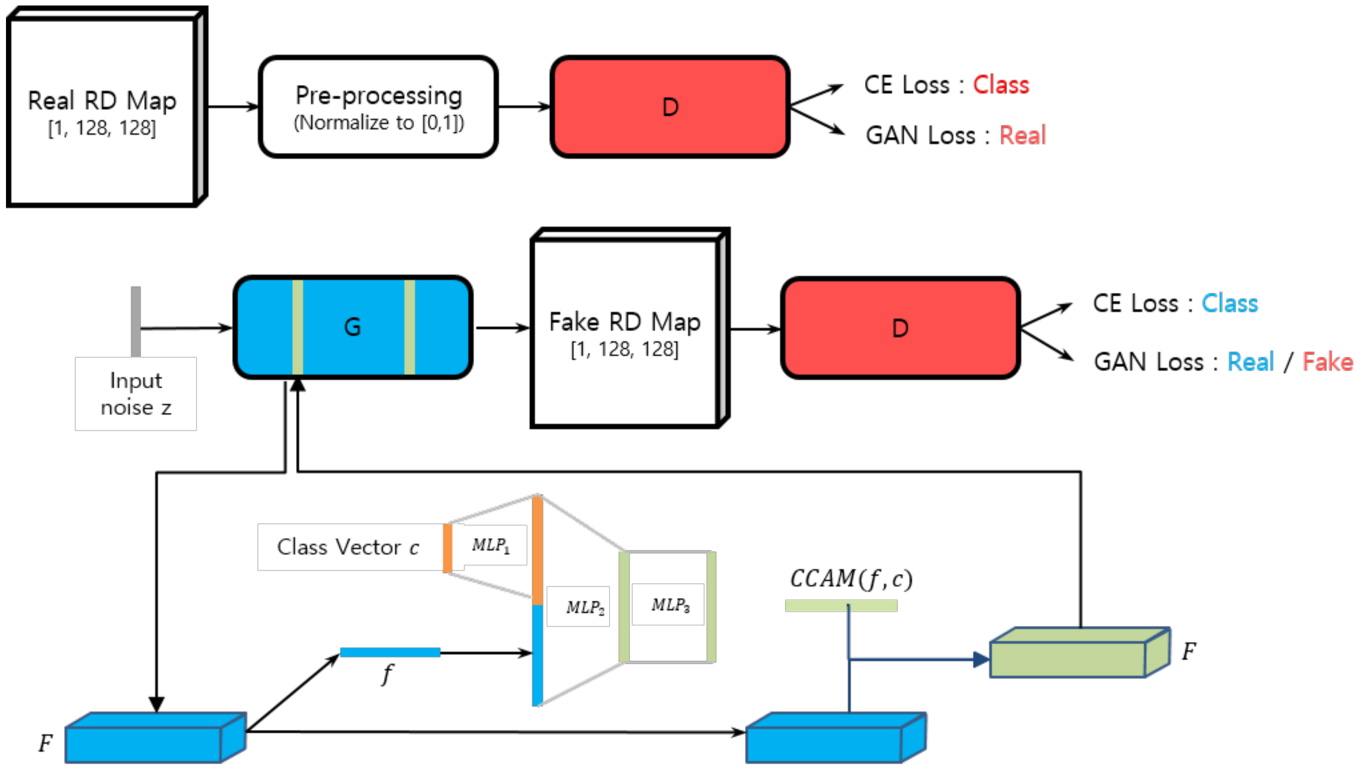


Fig. 1: The illustration of our conditional GAN training process. The loss with red font means that the discriminator is trained with that loss term, and the blue font means that the generator is updated with it. With real RD map, the discriminator learns to classify it as real, and also learns to classify its correct class. For fake RD map, the discriminator learns to classify it as fake, and the generator learns to classify it as fake, and the generator learns to make the discriminator to classify it correctly.

provides a meaningful improvement in the reliable training of our GAN, which leads to an important improvement in the performance of the UAV classifier, which we will introduce in the next section.

Second is the input RD map pre-processing method for the discriminator. An important property that RD maps differ from conventional image data is that they have a maximum boundary and a minimum boundary for the image data, such as [0, 1], [-1, 1] or [0, 255]. So generative models are designed to use $sigmoid$ or $tanh$ activations in the very last layer with upper and lower bounds. However, RD Maps have log scale values whose values can ideally be infinite, and this difference makes it easy for discriminators to distinguish between real and synthetic RD Map data, which can lead to training failures. We solved this problem by normalizing the RD map at the [0, 1] boundary by dividing each individual RD map by the **maximum** abstract value. We named this normalization as PIN, which is abbreviation of per-image normalization.

The overall Generator and Discriminator training procedure is depicted in Fig. 1.The lower part of the picture is the CCAM embedding part. The class one-hot vector is not input to the first layer, but instead to the two intermediate layers.

B. Synthetic data augmentation using conditional GAN

An important property of generative models is sampling. You can use a trained generator to sample z from a Gaussian normal distribution. If the model has learned the distribution of the real data well, it will generate an appropriate sample from it. Using our generator that we trained in the first phase, we can randomly sample RD maps $G(z|c = C)$ that match the given classes $C \in [0, 1, 2]$. The discriminator network is not deployed in this phase. All sampling processes are completed before the start of the classifier. That is, synthetic training images are not generated on-the-fly with classifier training. These class conditionally generated samples are then used for UAV classifier training described in Step 3.

C. UAV Classifier Training

Two situations were assumed in training the UAV classifier. The first is rich data setup. Here we use all training data to train a conditional GAN and UAV classifier. The second is the lack of data setting. About one-quarter of the original data is used to train conditional GAN and UAV classifiers, which severely degrades classification performance. The reason we retrain the GAN with fewer data is to comply with the restriction that the remaining 3/4 of the data is inaccessible.

For reliable training of image classifiers, it is common to normalize the training images to the global data mean and variance, but we do not apply it because the RD map dataset contains quite a few noisy samples that can shift the global mean and variance, whose deviations can have a bad effect. Therefore, we only use individually normalized real RD maps

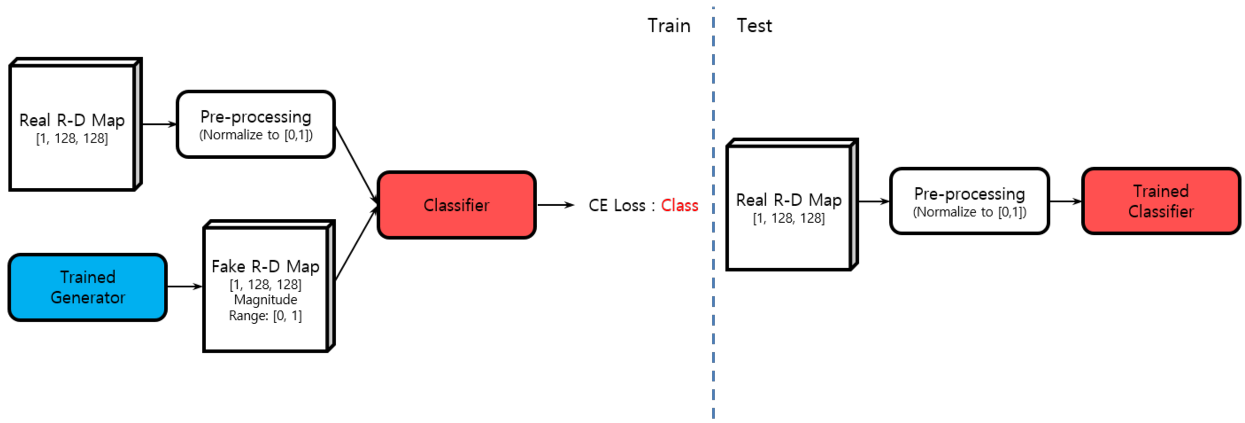


Fig. 2: The illustration of our UAV classifier training.

and generated fake RD maps. To avoid unwanted situations where the classifier is overparameterized or underparametered, it uses the exact same architectural network used for the discriminator in the GAN. If the conditional GAN training is stable, the classifier head of the discriminator will be able to classify the three classes of UAVS. This ultimately means that if the same kind of network is used for the classifier, the classifier's stability is also guaranteed. We have empirically confirmed that the classifier is sufficiently discriminatory. This is an important point for our data, as the three types of UAVs are actually completely indistinguishable from the human field of view. The training process of the classifier is rather simple compared to GAN training and is shown in Figure 2. By evaluating a trained classifier on a test data set, you can measure the accuracy gain of the classifier according to a data augmentation strategy.

IV. Experiment

For experiments, We collected 39,047 RD maps of 3 types of free-flying UAVs in total, and after randomly shuffling them, we divided the RD maps into two groups, 30,000 RD maps for train data and 9,047 RD maps for test data. We trained the conditional GAN based on the combinations of two criteria. The first criteria is the "The number of training data". Though we have 30,000 RD maps for training in total, we additionally wanted to assume the data-scarce environment, so we used only 10,000 of them, which may degrade the performance of the cGAN. Therefore, for the first criterion, we have **two setups** with 10,000 and 30,000 training data respectively. The second criteria is "The kind of cGAN models used". We previously introduced two training details that contributed to the stable training of GAN, which are per-image normalization (PIN), and conditioning channel attention module (CCAM). For the second criterion, we have **three setups**, cGAN, cGAN + PIN, cGAN + PIN + CCAM. Please aware that the cGAN does not mean the original form of vanila cGAN, but instead refers to the combination model of DCGAN, ACGAN, and LSGAN. With these two criteria, we trained totally 6 combination of cGAN models. For each setting, we also varied the number of synthetic (fake) samples that were sampled using the cGANs, and trained the corresponding UAV classifiers.

From now on, we verify that our training strategy stabilizes the training of GAN and also enhances the performance of the classifier, both quantitatively and qualitatively. We also provide the details of our training model and hyper-parameters that we used in our experiments.

A. *Quantitative Analysis*

Our model training setting that exploits the synthetic UAV RD map is actually a straight-forward way to measure the quality of the generated RD map. This measure was also recommended in CAS [25]. However, as CAS reported, the well-known conditional generative models in natural image domain suffered in actually enhancing the performance of the classifier, and the additional data even degraded the performance. On the other hand, our model showed decent improvements on the accuracy of UAV classifier, and the score is reported on the Table I.

The number of spurious samples used in the "None" column of the cGAN model is zero, meaning that the classifier does not use an additionally augmented RD map, so it is a vanilla setting for classifier training. As you can easily expect, the more training data you use, the higher the base accuracy of the UAV classifier.

In the third column, the performance degraded when using the synthetic RD map, a sample of the cGAN, for classifier training. This can also be predicted from the fact that the distribution of the real RD map and the fake RD map is different as mentioned in the sectionIII.

This issue was easily mitigated by using the PIN shown in the fourth column, cGAN + PIN. For most Real : Fake Ratios we could observe some performance improvement compared to vanilla classifier training. This means that synthetic RD maps were helpful.

Finally, the scores in the last column reflect that the fake RD map generated from the model using the CCAM embedding module was also helpful, and that using more sham samples improved the performance of the UAV classifier.

TABLE I: UAV classifier accuracy using the data augmentation of cGANs

Real : Fake Ratio	**cGAN Model**			
	None	**cGAN ([12] + [20] + [21])**	**cGAN + PIN**[a]	**cGAN + PIN + CCAM**
10000 : 0	69.82	-	-	-
10000 : 10000	-	62.01	69.67	70.12
10000 : 20000	-	63.19	70.98	73.50
10000 : 30000	-	63.88	70.76	**74.14** (+4.32)
30000 : 0	87.34	-	-	-
30000 : 10000	-	83.55	88.45	88.25
30000 : 30000	-	83.76	88.91	**90.91** (+3.57)

[a] As mentioned at the previous section, PIN is per-image normalization

B. Qualitative Analysis

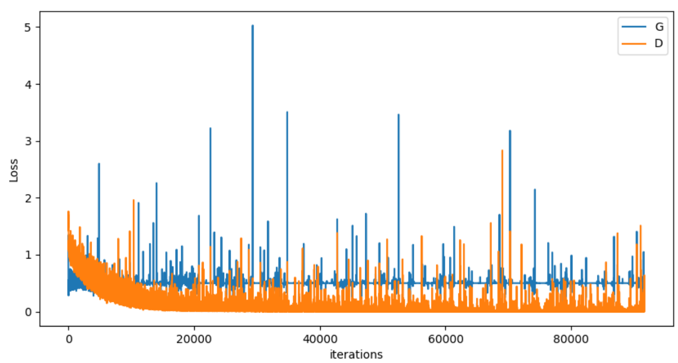


Fig. 3: The training loss curve of cGAN.

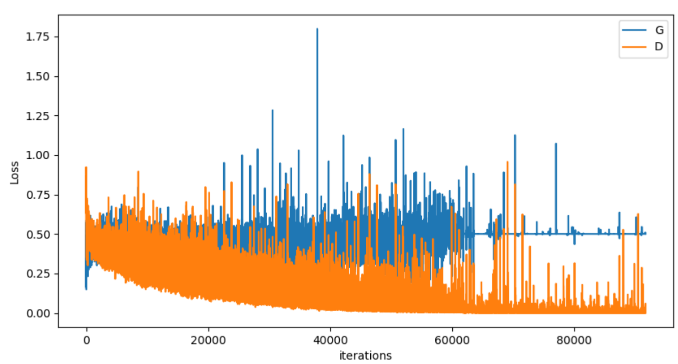


Fig. 4: The training loss curve of cGAN + PIN.

1) Loss Curves: We plotted the loss curve of the training of cGAN models at Fig 3~5. Please note that for the training of GAN, when either G loss or D loss converges fast, that phenomenon means that one of the network overwhelms the other side. This is harmful for the stable training, because the two networks should be trained competitively in order to learn useful information from each other, which is the fundamental motivation of GAN. For our model, since we adopt the loss of LSGAN, the ideal point of convergence is 0.25 for both G loss and D loss (assuming that the CE loss for UAV classification for both G and D converges to zero), and either being 0 or 0.5 means that one network beats the other.

At the cGAN case, in Fig 3., the training loss of D goes toward zero quickly, which means that D easily beats G, breaking the competitive aspects. When PIN is added, in Fig 4., the competitive aspects lasts longer, but eventually D beats G at about 60,000 iteration. When PIN and CCAM is added, in Fig 5., though the D loss goes toward 0 and G loss goes toward 0.5, the both loss curves fluctuate until the end of the training of the model.

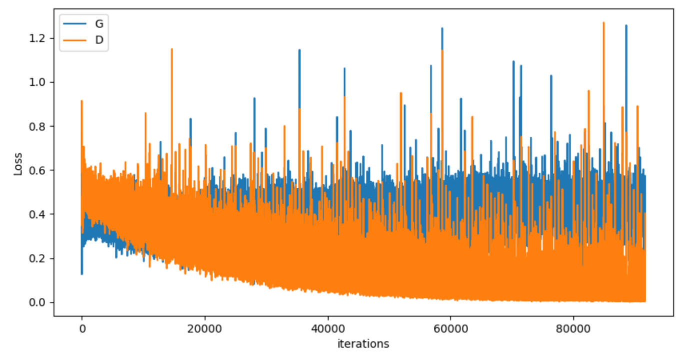
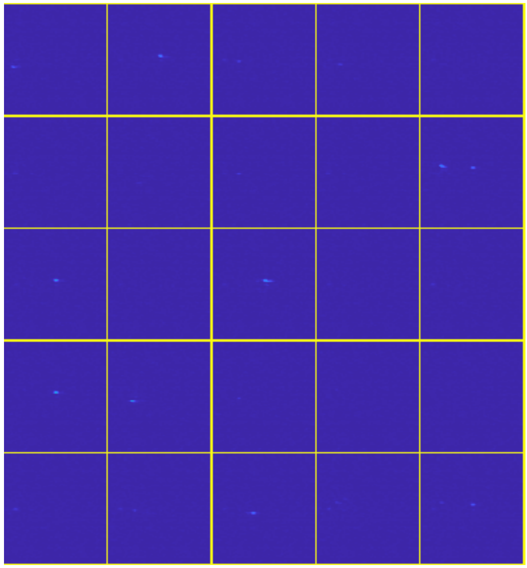
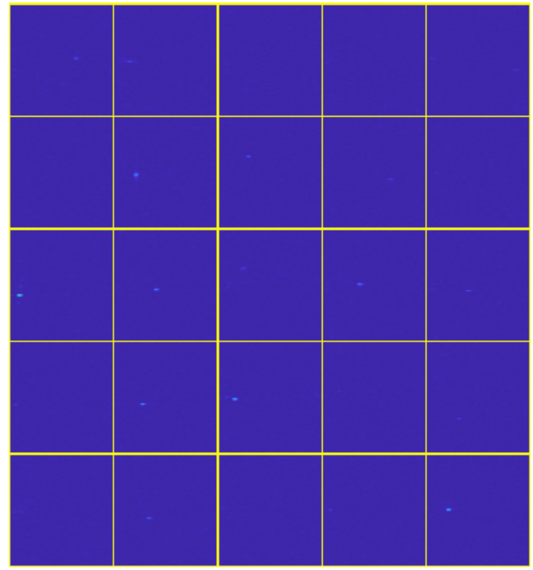


Fig. 5: The training loss curve of cGAN + PIN + CCAM.

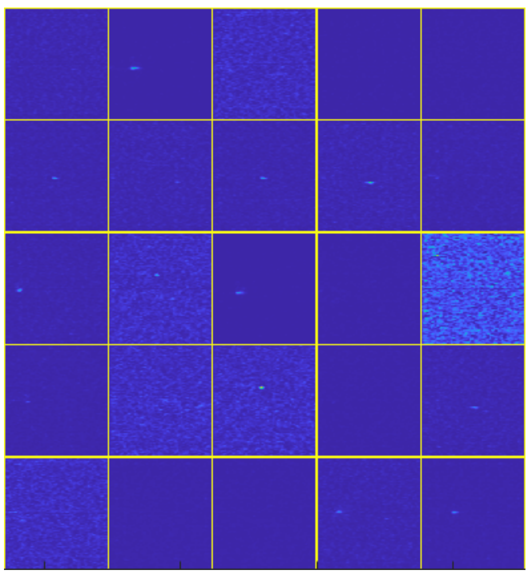
2) RD Map Visualization: To qualitatively evaluate the synthetic RD map, a randomly sampled synthetic RD map along with the actual training data was visualized in Figure 6. The input noise z as well as the class are randomly sampled. Although it is nearly impossible to discriminate the classes of RD maps in the human eye, the fact that class conditionally generated RD maps aid in the training of UAV classifiers supports that neural networks are capable of classifying UAVs. Note that the real RD maps used for training the models differ due to the PIN. As can be seen, since PIN force the maximum value of the signal to 1, it act as equalizer so that it becomes easy to find the peak signals. I also has drawbacks that when fine peak signal does not exist, it amplifies the noise, which can be harmful. Nevertheless, the previous experiments showed that it is superior. If we could eliminate every frame that radar failed to capture the UAVs, we expect that our method will perform even better. Also, the signals of the RD maps are stronger when cGAN + PIN + CCAM is used, and the base cGAN model produces many RD maps in which the peak signal appears to be non-existent. The performance degradation of cGANs and the according UAV classifiers may have occurred due to not filtering out the RD maps, which is a time-consuming task.



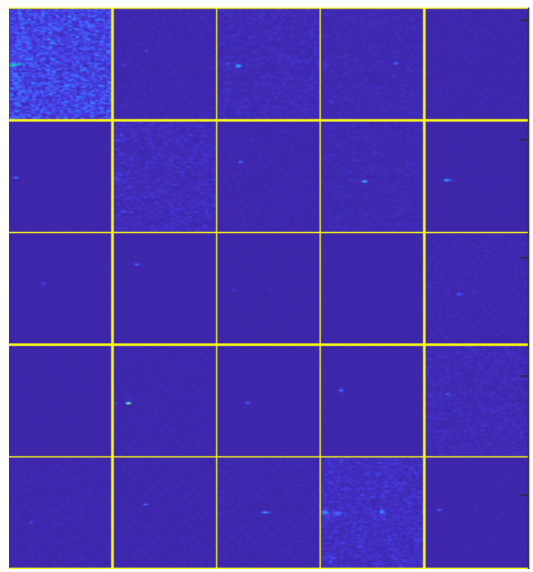
(a) cGAN: Fake



(b) cGAN: Real



(c) cGAN + PIN + CCAM: Fake



(d) cGAN + PIN + CCAM: Real

Fig. 6: Visualizations of synthetic RD maps (Fake) and real RD maps (Real) for cGAN model and cGAN + PIN + CCAM model. The G models for sampling these RD maps are trained using 30,000 of training data.

V. Conclusion

In this paper, we propose to use a conditional GAN model for augmenting RD map data of UAVs and train a UAV classifier using augmented RD maps, and demonstrate their effectiveness in both data-rich and data-poor settings. Larger performance gains in data-poor setups are of great benefit in that radar data is difficult to obtain. Two modifications were applied to reliably adopt the cGAN model used in previous vision work for radar tasks, contributing significantly to further improvements. Further performance improvement is expected if the frames which UAVs are non-existent are removed.

Acknowledgment

This work was supported by a grant-in-aid of HANWHA SYSTEMS.

References

[1] R. H. Rasshofer, M. Spies, and H. Spies, "Influences of weather phenomena on automotive laser radar systems" Advances in Radio Science, vol. 9, no. B. 2, pp. 49–60, July 2011.

[2] T. Wang, N. Zheng, J. Xin, and Z. Ma, "Integrating millimeter wave radar with a monocular vision sensor for on-road obstacle detection applications," Sensors, vol. 11, no. 9, pp. 8992–9008, Sept. 2011.

[3] D. A. Dolgov and C. P. Urmson, "Modifying behavior of autonomous vehicles based on sensor blind spots and limitations," June 2016, uS Patent 9,367,065.

[4] Abdulatif, S., Armanious, K., Aziz, F., Schneider, U., Yang, B. (2019, September). "Towards adversarial denoising of radar micro-Doppler signatures". In 2019 International Radar Conference (RADAR) (pp. 1-6). IEEE.

[5] Armanious, K., Abdulatif, S., Aziz, F., Schneider, U., Yang, B. (2019, September). "An adversarial super-resolution remedy for radar design trade-offs." In 2019 27th European Signal Processing Conference (EUSIPCO) (pp. 1-5). IEEE.

[6] Park, D., Lee, S., Park, S., Kwak, N. (2021). "Radar-spectrogram-based UAV classification using convolutional neural networks." Sensors, 21(1), 210.

[7] De Oliveira, M. L. L., Bekooij, M. J. (2020, September)." Generating Synthetic Short-Range FMCW Range-Doppler Maps Using Generative Adversarial Networks and Deep Convolutional Autoencoders." In 2020 IEEE Radar Conference (RadarConf20) (pp. 1-6). IEEE.

[8] B. Dekker, S. Jacobs, A. S. Kossen, M. C. Kruithof, and A. G. Huizing, "Gesture Recognition with a Low Power FMCW Radar and a Deep Convolutional Neural Network," in 14th European Radar Conference, 2017, pp. 163–166.

[9] B. Jokanovic and M. Amin, "Fall Detection Using Deep Learning in Range-Doppler Radars," IEEE Transactions on Aerospace and Electronic Systems, vol. 54, no. 1, pp. 180–189, 2018.

[10] J. M. Garc ́ıa, S. Member, and D. Zoeke, "MIMO-FMCW Radar-Based Parking Monitoring Application With a Modified Convolutional Neural Network With Spatial Priors," IEEE Access, vol. 6, pp. 41 391–41 398, 2018.

[11] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 115(3):211–252, 2015.

[12] Radford, A., Metz, L., Chintala, S. (2015). "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv:1511.06434.

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

[14] Mirza, M., Osindero, S. (2014). "Conditional generative adversarial nets." arXiv preprint arXiv:1411.1784.

[15] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, ́ Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. "Photorealistic single image super-resolution using a generative adversarial network." In CVPR, volume 2, page 4, 2017.

[16] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. "Learning to discover cross-domain relations with generative adversarial networks." arXiv preprint arXiv:1703.05192, 2017.

[17] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A. A. (2016). "Context encoders: Feature learning by inpainting." In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2536-2544).

[18] Chung, I., Park, S., Kim, J., Kwak, N. (2020, November). "Feature-map-level online adversarial knowledge distillation." In International Conference on Machine Learning (pp. 2006-2015). PMLR.

[19] Yoo, Y., Park, S., Choi, J., Yun, S., Kwak, N. (2017). "Butterfly effect: Bidirectional control of classification performance by small additive perturbation." arXiv preprint arXiv:1711.09681.

[20] Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Paul Smolley, S. "Least squares generative adversarial networks." In Proceedings of the IEEE International Conference on Computer Vision, pp. 2794–2802, 2017.

[21] Odena, A., Olah, C., Shlens, J. (2017, July). "Conditional image synthesis with auxiliary classifier gans." In International conference on machine learning (pp. 2642-2651). PMLR.

[22] Choi, Y., Choi, M., Kim, M., Ha, J. W., Kim, S., Choo, J. (2018). "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation." In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8789-8797).

[23] Chang, S., Park, S., Yang, J., Kwak, N. (2019). "Sym-parameterized dynamic inference for mixed-domain image translation." In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 4803-4811).

[24] Erol, B., Gurbuz, S. Z., Amin, M. G. (2019, April). GAN-based synthetic radar micro-Doppler augmentations for improved human activity recognition. In 2019 IEEE Radar Conference (RadarConf) (pp. 1-5). IEEE.

[25] Ravuri, S., Vinyals, O. (2019). Classification accuracy score for conditional generative models. arXiv preprint arXiv:1905.10887.