IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 31, 2018, accepted March 4, 2018, date of publication March 19, 2018, date of current version April 18, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2815741

Digital Signal Modulation Classification With Data Augmentation Using Generative Adversarial Nets in Cognitive Radio Networks

BIN TANG[1], YA TU[2], ZHAOYUE ZHANG[3], AND YUN LIN[2]
[1]College of Shipbuilding Engineering, Harbin Engineering University, Harbin 150001, China
[2]College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China
[3]College of Air Traffic Management, Civil Aviation University of China, Tianjin 300300, China

Corresponding author: Yun Lin (linyun@hrbeu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61771154.

ABSTRACT Automated modulation classification plays a very important part in cognitive radio networks. Deep learning is also a powerful tool that we could not overlook its potential in addressing signal modulation recognition problem. In our last work, we propose a new data conversion algorithm in order to gain a better classification accuracy of communication signal modulation, but we still believe that the convolution neural network (CNN) can work better. However, its application to signal modulation recognition is often hampered by insufficient data and overfitting. Here, we propose a smart approach to programmatic data augmentation method by using the auxiliary classifier generative adversarial networks (ACGANs). The famous CNN model, AlexNet, has been utilized to be the classifier and ACGAN to be the generator, which will enlarge our data set. In order to alleviate the common issues in the traditional generative adversarial nets training, such as discriminator overfitting, generator disconverge, and mode collapse, we apply several training tricks in our training. With the result on original data set as our baseline, we will evaluate our result on enlarged data set to validate the ACGAN's performance. The result shows that we can gain 0.1∼6% increase in the classification accuracy in the ACGAN-based data set.

INDEX TERMS Cognitive radio, modulation recognition, pattern recognition, classification algorithms, deep learning, convolutional networks, generative adversarial net.

I. INTRODUCTION

Nowadays, wireless communication features a settled spectrum assignment policy. With the boosting requirements for wireless band-width of radio spectrum, it's important to exploit the existing wireless spectrum opportunistically [1]. Without effective measures to regularize wireless spectrum, wireless network will face the serious threat of jamming signals [2]. Hence, people pay more and more attention to improving opportunistic spectrum access techniques, acknowledged as Cognitive Radio (CR) [3], [4]. To adjust its parameters to strengthen its ability to reliably communicate, CR are radios that are able to learn their surrounding environment. Currently, wireless spectrum demand is putting spurs to the improved radio efficiency [5], [6]. CR is proposed as one method to be a more open spectrum policy. The famous spectrum assignment policy strategy in CR is Dynamic Spectrum Access (DSA). In DSA, radios would detect whether licensed spectrum is available for use [5]. If the cognitive radio does not find the primary user of channel, then this radio is free for a secondary user to choose the channel. It is well known that radio interference will not only have a negative effect on communication, but also causes issue on physical layer security [6]. Modulation Recognition is a way for cognitive radio to detect this appearance to avoid radio inference and optimize spectrum allocation [7], [8]. Therefore, a smart approach should be found to recognize the signal modulation.

Deep Learning (DL) is an emerging field of Machine Learning (ML) method thanks to big data [9] and the improvement in hardware [10]. DL has recently gained attention because of the successful applications in Computer Vision (CV), Natural Language Processing (NLP) and so on. Researchers are enthusiastic trying to extend DL to other domains, including signal modulation recognition. Compared to traditional ML, DL can achieve further performance improvements, since DL has a flexible mechanism to extract feature itself to optimize the parameter to get a better end to end performance [11]. Though, DL has so many merits, its downsides should not be forgotten. Without the big data support, DL model are prone to overfit for its deep and complex structure [11]. Convolutional Neural Networks (CNN) is the main framework of DL in CV. There have been several works researching on the application of DL in Automated Modulation Classification (AMC), paper [5] proposes by using Spectral Correlation Function (SCF) pattern and Deep Belief Network (DBN) to develop AMC. Paper [12] addresses the issues of AMC in CR by performing classification on complex CNN. To fully exploit the capacity of CNN, Yao YuDong and his students implements famous CNN model, AlexNet and GoogLeNet, with 3-channel images in modulation recognition [13]. It really obtains good accuracy. In our work, we propose a new data conversion algorithm that makes CNN significantly perform better. What's more, it's a promising job to explore modulation recognition based on DL's potential in addressing the radio inference and jamming issues descripted in paper [14].

In this paper, we will make full use of CNN in signal modulation recognition and enlarge our dataset by implementing data augmentation with GAN. GAN are put forward by Goodfellow *et al.* [15]. This technique is one of unsupervised machine learning artificial intelligence algorithms which can be applied to generating images that look superficially authentic to human observers. This framework consists of a mechanism of two neural networks contesting with its counterpart in a cat-mouse game. GAN has been utilized in various applications, such as high-resolution image generation [16], interactive image generation [17], and image inpainting [18], we also want to exploit its potential in signal modulation recognition. In this paper, we focus on the real fact that our labeled data is limited and we are attempting to find out how to use GAN to generate images that may help enlarge original dataset effectively by data augmentation.

When it comes to DL, plentiful high-quality data is the key to good DL model. Data augmentation is often used to expand the training dataset [19]. The prevalent data augmentation is generating additional data through geometric transformations, such as translation, rotation, scaling, randomly cropping, from existing data [20]. However, these approaches are from a priori known invariance and only lead to an image-level transformation through depth and scale. GAN will attempt to learn the distribution which represents the high-level feature of real data [20]. In this paper, we enlarge our dataset by generating images while remain high-level features extracted from original images.

The rest of this paper is presented as follow. Section II provides an overview of how to obtain the dataset in this paper. Section III will offer a brief introduction to GAN basic theory and Section IV will give the GAN training tricks we implement to combat with the common training issues and the result about performance comparison between AlexNet trained on original dataset and AlexNet trained on GAN-based data augmentation dataset.

II. CONTOUR STELLA IMAGE DATASET

In our related work, we make full use of the CNN model in signal modulation recognition. Our contribution is proposing a new data conversion algorithm that makes CNN significantly outperform the same CNN model mentioned in the contrast paper [13]. Since CNN accept image data format, the contrast paper [13] chooses constellation diagram to feed with CNN. By contrasting the result of CNN (AlexNet, GoogLeNet) between different images dataset, the contrast paper [13] proves that gray images are only one channel is utilized and the capacity of these models is not fully exploited, so the result will be better if we use colored image. However, they still use almost the same color (mainly dominated by black and blue colors) in the image. In order to get a better training performance, our paper proposes a new method to make the colored image training set more discernible.

Our data conversion algorithm will convert Constellation Diagram to another image format, which we define it as Contour Stellar Image. Compared with Constellation Diagram, Contour Stellar Image gets more color feature. Constellation Diagram has been widely applied to representing a modulated signal by using its sample points' amplitude and phase information. The sample points from modulated signal, which is polluted by noise, won't get gather in one point in Constellation Diagram for its amplitude and phase information has been disturbed. It also means that different area in Constellation Diagram has different density of sample point. The main idea of our new algorithm is to take full advantage of the dots' destiny to let the dataset be more discernible.

To produce Contour Stellar Image, the first thing is to convert complex signal into Constellation Diagram. Taking QPSK at SNR = 4dB as an example, this phase-mapping is shown in (1):

$$s(t_i) = e^{j2\pi f_c t_i + \pi\frac{2c_i+1}{4}} + n(t_i), \quad c_i \in 0, 1, 2, 3 \tag{1}$$

Where t_i is time node, $n(t_i)$ is noise, f_c is carrier wave's frequency and c_i is the signal phase.

Then select a small square window function which will slide on Constellation Diagram. We define the square window function as Destiny Window Function. Sliding on Constellation Diagram, Destiny Window Function will sum how many signal sample point in the current area. Different result will be mapped to different signal sample point density and different color. After converting the raw QPSK at 4dB into Constellation Diagram, the QPSK at 4dB Constellation Diagram is shown in Fig. 1:

In Fig. 1, the red square (Density Window Function) will count how many dots in its field. After calculating every area's Relatively Point Destiny by using (2), we will map the area's Relatively Point Destiny into a color bar based on its normalized value, where Yellow means higher density area, Green means middle density area and Blue means lower density area. Fig. 2 shows the color bar we choose.

Even disturbed by the noise, the sample point in QPSK Constellation Diagram at SNR = 4dB is still more likely

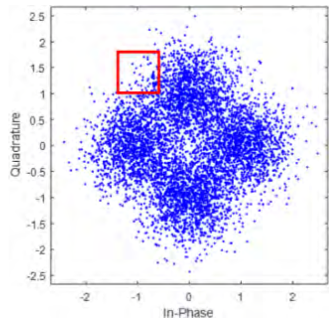

FIGURE 1. QPSK constellation diagram at SNR=4dB.

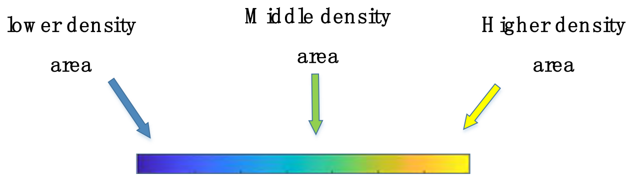

FIGURE 2. Relatively point destiny colorbar.

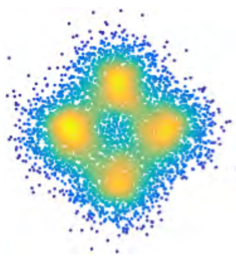

FIGURE 3. QPSK Contour Stella image at SNR=4dB.

to gather in the middle of clusters. That means the center of the cluster will present as Yellow. Implemented with our new data conversion algorithm, QPSK Constellation Diagram at SNR = 4dB will be converted to Fig. 3:

This process can be related to (2):

$$\rho(i,j) = \frac{\sum_{i=x_1}^{x_2} \sum_{j=y_1}^{y_2} dots(i,j)}{\sum_{x1=W_0}^{W_1} \sum_{y1=H_0}^{H_1} \sum_{i=x_1}^{x_2} \sum_{j=y_1}^{y_2} dots(i,j)} \tag{2}$$

Where x_1, y_1 stands for top-left corner of Destiny Windows Function currently coordinate, x_2, y_2 stands for bottom-right corner of Destiny Windows Function currently coordinate, W_0, H_0 stands for Constellation Diagram top-left corner coordinate, W_1, H_1 stands for bottom-right corner coordinate of Constellation Diagram, $\rho(i,j)$ is Relatively Point Destiny, and $dots(i,j)$ means one sample point, whose coordinate is (i,j) in Constellation Diagram.

To have a better comparison to the contrast paper [13], we select the same category modulated signals: BPSK, 4ASK, QPSK, OQPSK, 8PSK, 16QAM, 32QAM, 64QAM. Through the process mentioned above, the Contour Stellar Image of at the SNR of 14dB, 4dB will be presented as Fig. 4 (a) and (b):

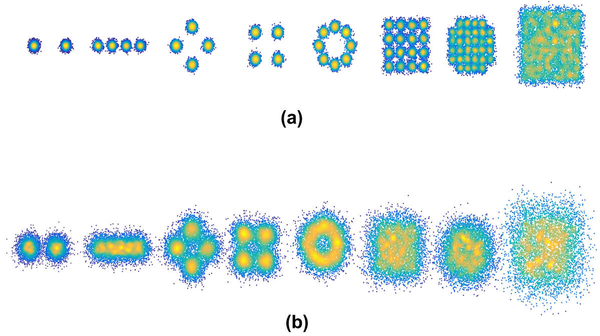

FIGURE 4. Contour Stellar images of 8 signals at different SNR. The class of the modulated signal in Fig. 4 is: BPSK, 4ASK, QPSK, OQPSK, 16QAM, 32QAM, 64QAM. (a) SNR = 14dB. (b) SNR = 4dB.

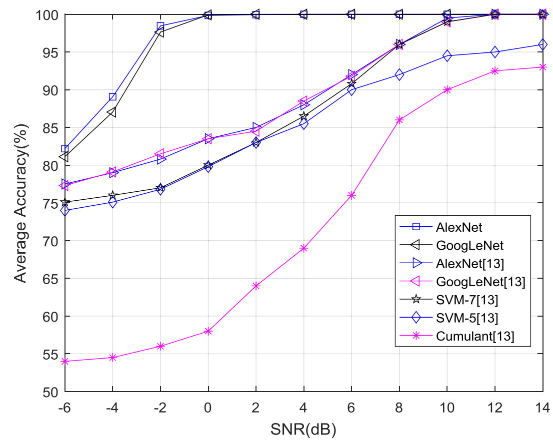

FIGURE 5. Average accuracy versus SNR.

Seen from the Fig. 4, it's obviously that Contour Stellar Image obtain more features in color. It means CNN training can not only count on the shape of Contour Stella Image, but also count on the color in it. That's the reason makes the dataset more discernible.

We use MATLAB 2017a to generate the training and test dataset we need. Being different to the contrast paper [13] use 100000 images to train, we only use 10000 images to train. Our dataset consists of the same 8 category modulated signals, which is BPSK, 4ASK, QPSK, OQPSK, 8PSK, 16QAM, 32QAM, 64QAM.The parameter of every signal is equal to the contrast paper [13]. Every signal will be disturbed 10000 contour stellar images for training set, 1000 for test set. The SNR will range from -6dB from 14dB and the stride is 2dB. Hyper parameter is also tuned to reduce the training time while keeping the accuracy.

As shown in Fig. 5, Fig. 6 and Fig. 7, following conclusions can be drawn:

1. The DL trained on Contour Stellar Image dataset performance are much better than the DL performance in the article [13] we want to compare with.

2. With the support of big data, DL framework outperforms the traditional ML.

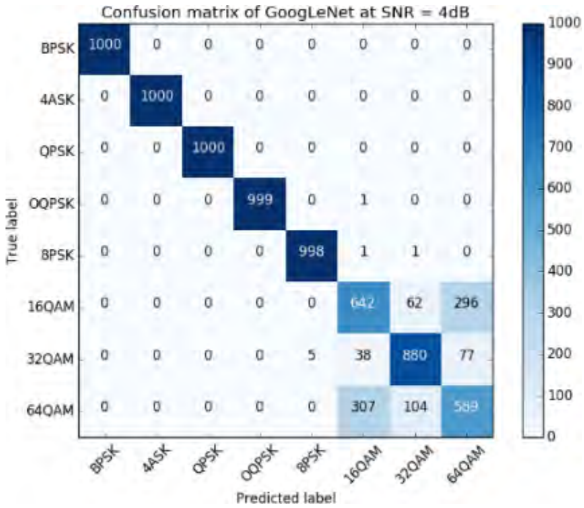

FIGURE 6. GoogLeNet [13] classification matrix at 4dB.

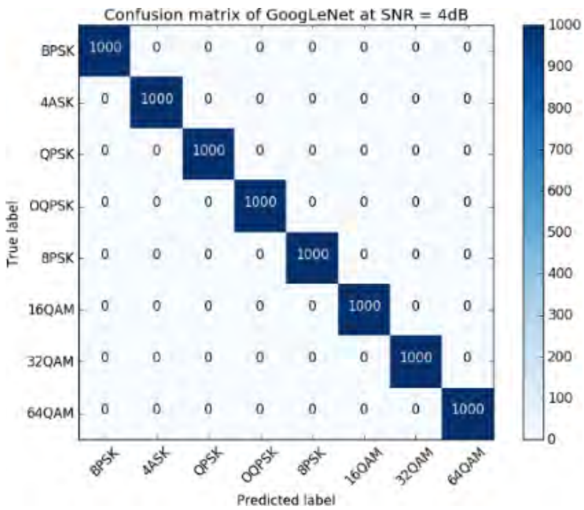

FIGURE 7. GoogLeNet trained on Contour Stella image classification matrix at 4dB.

3. Our work just used 10000 (versus 100000 in contrast paper [13]) training dataset to get better result. That means the attribute of the dataset plays an important role in DL classification task.

III. GENERATIVE ADVERSARIAL NETS

GAN was introduced by Goodfollow *et al.* [15]. GAN can be used to generate images from an adversarial training. There are two basic components named generator and discriminator in GAN. The generator will be trained to map from a latent space to a specified distribution which is close to real data distribution. The discriminator will attempt to distinguish the real data distribution from the fake data distribution, whose process is averse to generator. This process can be treated as a competitive setting. In CV, the generator is a deconvolutional neural network while the discriminator is a convolutional neural network. This model can be depicted in Fig. 8, when we feed GAN with a OQPSK at 4dB SNR image:

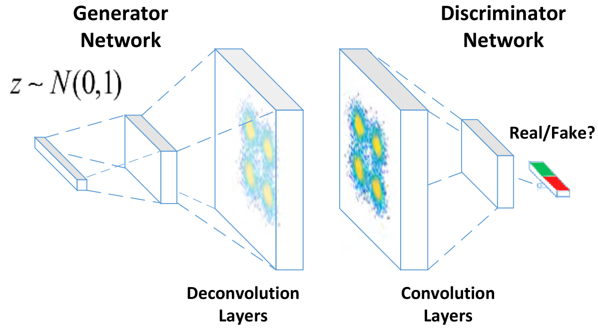

FIGURE 8. GAN's training.

In Fig. 8, generator generates a fake OQPSK at 4dB SNR image which is very similar to the real OQPSK at 4dB SNR image, both real image and fake image will be sent into discriminator. The discriminator has to tune its parameter to tell the true result that the real image is from Contour Stellar Image dataset and fake image is from generator. Generator will adjust its parameter to make the fake OQPSK look more like to the one picked from Contour Stellar Image, if the discriminator gives the true answer. Discriminator and generator are trained adversarially to improve by competing with each other. The traditional GAN losses in this cat-mouse game can be related to:

$$\min_G \max_D V(D, G) = E_{x\sim P_{data(x)}}[\log D(x)] + E_{z\sim P_{z(z)}}[\log(1 - D(G(z)))] \quad (3)$$

Where $V(G, D)$ is the combined loss for the GAN, $D(x)$ is the result of discriminator to predict the real image, $G(z)$ stands for the fake image comes from generator and $D(G(z))$ means the result of discriminator to predict the fake image.

In the generator phase of training, the best result for generator is $D(G(z)) = 1$, it implies discriminator has been fooled to predict all the fake image to real image. To generator, the optimization problem is:

$$\min_G V(D, G) = \min_G (E_{z\sim Pz(z)}[\log(1 - D(G(z)))]) \quad (4)$$

Then the generator will update weight by descending its gradient:

$$\nabla\theta_g \frac{1}{n}\sum_{i=1}^{n} \log\left\{1 - D\left[G(z^{(i)})\right]\right\} \quad (5)$$

Where n is training data batch size, $z^{(i)}$ is the noise vector in the minibatch.

On the other hand, the discriminator tries to maximize (3). The best result for discriminator is $D(x) = 1$ and $D(G(z)) = 0$. To discriminator, the optimization problem is:

$$\max_D V(D, G) = \max_D (E_{x\sim P_{data}}[\log(D(x)] + E_{z\sim P_{z(z)}}[\log(1 - D(G(z)))]) \quad (6)$$

Then we will update the generator's weight by descending its gradient:

$$\nabla\theta_d \frac{1}{n}\sum_{i=1}^{n}[\log D(x^{(i)}) + \log(1 - D(G(z^{(i)})))] \tag{7}$$

When generator fixed, (6) can be reted as:

$$\begin{aligned} V(G,D) &= \int_x p_{data}(x)\log(D(x))dx \\ &\quad + \int_z p_z(z)\log(1 - D(g(z)))dz \\ &= \int_x p_{data}(x)\log(D(x)) \\ &\quad + p_g(x)\log(1 - D(x))dx \end{aligned} \tag{8}$$

When it comes to the function:

$$f(x) = m\log(x) + n\log(1 - x) \tag{9}$$

(9) reaches maximum in [0, 1] at a $\frac{m}{m+n}$, the optimal discriminator D is:

$$D_G = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \tag{10}$$

At this time, (3) can be reted as:

$$\begin{aligned} \max_D V(G,D) &= E_{x\sim P_{data}}[\log D_G(x)] \\ &\quad + E_{z\sim P_z}[\log(1 - D_G(G(z)))] \\ &= E_{x\sim P_{data}}[\log D_G(x)] \\ &\quad + E_{z\sim P_z}[\log[(1 - D_G(x))] \\ &= E_{x\sim P_{data}}[\log \frac{P_{data}(x)}{P_{data}(x) + P_g(x)}] \\ &\quad + E_{z\sim P_z}[\log \frac{P_g(x)}{P_{data}(x) + P_g(x)}] \end{aligned} \tag{11}$$

This competition will stop when discriminator doesn't know whether the image sent to it is real or fake and generator can't improve the quality of image, then they reach the stage named "Nash Equilibrium". At this time, D(x) = 0.5, $p_{data}(x) = p_g(x)$, it implies discriminator will randomly give the prediction result about the image sent to it and (11) will be:

$$\begin{aligned} \max_D V(G,D) &= E_{x\sim P_{data}}[\log D_G(x)] \\ &\quad + E_{z\sim P_z}[\log(1 - D_G(G(z)))] \\ &= E_{x\sim P_{data}}[\log D_G(x)] \\ &\quad + E_{z\sim P_z}[\log[(1 - D_G(x))] \\ &= E_{x\sim P_{data}}[\log \frac{P_{data}(x)}{P_{data}(x) + P_g(x)}] \\ &\quad + E_{z\sim P_z}[\log \frac{P_g(x)}{P_{data}(x) + P_g(x)}] \\ &= E_{x\sim P_{data}}[\log \frac{1}{2}] + E_{z\sim P_z}[\log \frac{1}{2}] \\ &= -\log(4) \end{aligned} \tag{12}$$

In summary, the global minimum of $V(G, D)$ is achieved if and only if $p_{data}(x) = p_g(x)$. At that time, $V(G, D)$ achieves the value $-\log 4$.

However, there are some flaws that couldn't be ignored in traditional GAN framework:

1. Since the cost functions of GAN are non-convex, the parameters are continuous, and the parameter space is extremely high-dimensional, so gradient descent is easy to disconverge [21].

2. Traditional GAN can't generate one specified class image

3. GAN model often gets trapped in mode collapse [21], since generator always emits the same point and discriminator has no ability to check the other sample in the minibatch.

IV. IMPROVED DL SIGNAL MODULATION RECOGNITION FRAMEWORK BASED ON ACGAN

In this section, we proposed a new DL framework in signal modulation. Taking the CR's reality into consideration, the real data is limited. We couldn't get as many data as we wish. However, as it's mentioned in Section I, DL tend to get overfitting without enough data, due to the fact that insufficient data will lead to classification decision boundary [11]. In order to make full use of DL capacity in signal modulation recognition, we choose the generative model to achieve the techniques of data augmentation.

In our experiment, traditional GAN seems more likely to fail to converge, since there is no control on process the data being generated. Generator will fool discriminator with garbage images. To direct the data generation process, we implement data augmentation with CGAN [22] (Conditional GAN). In CGAN, the conditional information could be any type of auxiliary information, for example, data class label. In order to perform the conditioning, we could concatenate the conditional information and data as the input to generator and discriminator. CGAN can be deployed to generate conditional results. That means we could use CGAN to generate the one certain class images when we feed a certain class label. The CGAN can be related to (13) [22]:

$$\begin{aligned} \min_G \max_D V(D,G) &= E_{x\sim P_{data(x)}}[\log D(x,y)] \\ &\quad + E_{z\sim P_{z(z)}}[\log(1 - D(G(z,y)))] \end{aligned} \tag{13}$$

Where x is the real data distribution, y is the auxiliary information, z is the noise.

ACGAN [23] is an outstanding representation of CGAN. Paper [23] compare their work with paper [21], the result in MS-SSIM proves their generated images are diversity and discriminable. Diversity dataset will help CNN find out which feature really matters [19]. For the purpose of effective data augmentation, in this paper, ACGAN is used to realize data augmentation in Contour Stellar Image. ACGAN is not tremendously different from CGAN, it will use the combinate information, corresponding class label c and noise z, as the input to generator. Generator will generate fake images G(c, z) to cheat discriminator. The discriminator will give

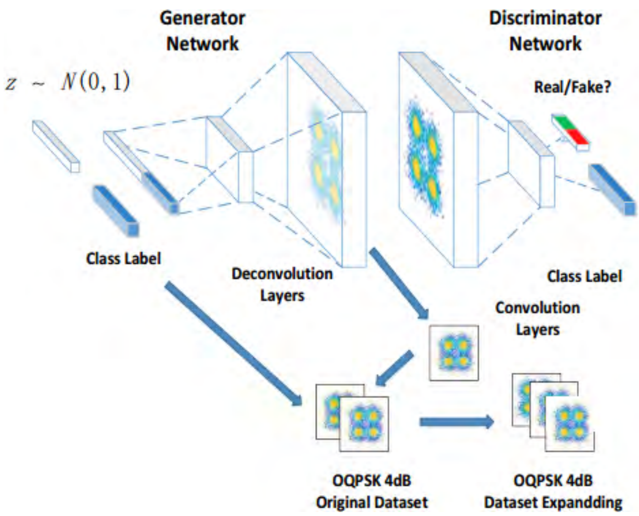

FIGURE 9. An illustration of the proposed framework of using ACGAN for data augmentation.

a prediction on the source of image and the class it should belongs to. The Loss function has two parts: the log likelihood of the correct source, Ls, and the log likelihood of the correct class, Lc. This model can be related to (14), (15) [23]:

$$Ls = E[\log P(S = real|X_{real})] + E[\log P(S = fake|X_{fake}))] \tag{14}$$

$$Lc = E[\log P(C = c|X_{real})] + E[\log P(C = c|X_{fake}))] \tag{15}$$

Generator is trained to maximize Lc − Ls, while discriminator is trained to maximize Lc + Ls. ACGAN learns the representation related to noise and is independent of class label. In paper [23], they prove this modification to the standard GAN provides excellent results and stabilize training. In original GAN [15] learns a mapping from uncontrolled low-dimensional manifold to high-dimensional data spaces, while ACGAN using the class label to control this process. By using ACGAN, we will form a plentiful and stable feature-level manifold thanks to the generated samples.

Fig. 9 represents the framework of using ACGAN for data augmentation:

In Fig. 9, we firstly choose the class label to decide which kind of Contour Stellar Image we want to generate. For example, we want to generate OQPSK at 4dB SNR, we give generator the corresponding label. With the combination with label and noise, generator will generate fake OQPSK at 4dB to fool discriminator. Discriminator will give a prediction about source of this image and class label it should have. After the training, this game should get to the stage named "Nash Equilibrium" and the accuracy of auxiliary classifier in discriminator can't improve more. Then we feed the generator with the class label and correspond Contour Stellar Image will be produced, which will be sent to the corresponding class in the dataset.

Here, we proposed a new framework for applying CNN in signal modulation recognition. This whole framework can be illustrated by Fig. 10.

Firstly, converting complex signal received from radio into Constellation Diagram, then using the new data conversion algorithm to convert Constellation Diagram into Contour Stella Image, which will be our training dataset. After we store enough modulated signal data, we train ACGAN to implement data augmentation. Then both generated images and real images will be sent to CNN for the purpose of training CNN. After the training, the weights in CNN will be reused to predict what type of modulated signal we are receiving.

V. EXPERIMENT

As has been talked in paper [21], GAN's stabilize training is an ongoing research topic. The classifier in our framework is AlexNet and this model need high resolution picture as input data. So, we select the ImageNet Hyper parameter mentioned in the paper in [23] that our ACGAN could be capable of generating high resolution picture. But even we use the hyper parameter suggested in the paper [23], we still observe some unexpected phenomenon, such as discriminator overfitting, mode collapse and generator disconverge in our dataset. To obstacle those issues, we introduce several tricks in GAN training:

1. Normalized the input images between −1 and 1. This trick can be related to (16):

$$x' = \frac{x - mean(x)}{\max(x) - \min(x)} \tag{16}$$

Where x is the image data, x' is normalized data.

This trick can accelerate training and prevent overfitting.

2. Add the Gaussian Noise to the input image. The problem is the overfitting of discriminator. One of the reasons leading to this is discriminator may notice that images from true distribution is a matrix of numbers of the form n/255. So, adding gaussian noise to the input images will blur this fine detail, which helps to avoid the problem.

3. In paper [24], spherical linear interpolation will produce sharper samples than linear interpolation does. So, we choose to interpolate a gaussian distribution rather than a uniform distribution for noise.

4. In traditional GAN training method, the minibatch sent to discriminator consists of real images and fake images. We change the minibatch component which only contains all real images or all generated images. This will make gradient descent in discriminator smoother and controllable. Fig. 11 depicts this process:

5. Label smoothing and Noisy. According to paper [21], replaces the 0 and 1 targets for a classifier with smoothed values like 0.1 and 0.9 will make the discriminator work harder. Without label smoothing, the discriminator is extremely liable to overfitting, In that case, GAN model much better objective function to target. In (17) depicts the loss function:

$$d_{\sigma,JS}(q_\theta||p) = JS[p_\sigma * q_\theta||p_\sigma * p] \tag{17}$$

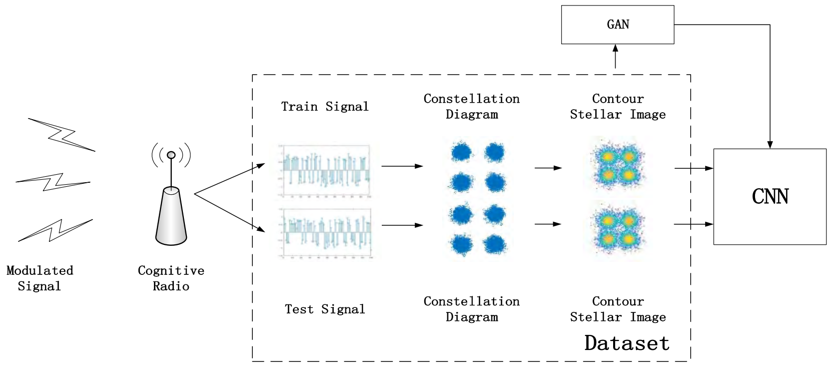

FIGURE 10. **DL in signal modulation recognition.**

FIGURE 11. **Training minibatch component.**

Where θ is the parameter of GAN, p_θ is the generated image distribution, p is the real data image distribution, $*$ is convolution operation, p_σ is noise, and JS is Jensen-Shannon divergence [15].

In (17), the fine part of the data distribution is blurred out and it will make GAN model less sensitive to fine patterns.

In our experiment, we choose to replace the real label with a random number between 0.7 and 1.2. Replace with random number between 0.0 and 0.3 for the fake image label. Expect the soft label, training label will be occasionally flipped for discriminator, which means the random number between 0.7 and 1.2 will be the fake images' label and the random number between 0 and 0.3 will be real images' label. It indeed prevents discriminator from overfitting.

6. Train discriminator more. Discriminator should be ahead of the generator so that it could lead generator. What's more, discriminator should be trained more to get a better classification accuracy since we add GuassianNoise in the input image.

7. Using Minibatch Discrimination Layer. In paper [21], the common failure mode for generator is mode collapse. In GAN, discriminator checks each example independently. Generator can win discriminator by fool discriminator the same image which is very similar to the real image. That will lead to no diversity in the generate image and no data augmentation. Fortunately, paper propose a new technique named Minibatch Discrimination that enables discriminator see different sample in the minibatch so that discriminator has the capacity to separate the identical outputs. In (18), (19) depicts the process:

$$d(c_i||c_j) = \exp(-||c_i - c_j||_{L_1}) \tag{18}$$

$$D(c_i) = \sum_{j=1}^{n} d(c_i||c_j) \tag{19}$$

Where c_i is the current sample that discriminator checks, c_j is other samples in the minibatch, n is the amount of sample.

After obtaining the (19) result, the result will be concatenated to the output of c_i, so discriminator gets additional information to prevent generator from fooling it only with one image.

8. Virtual Batch Normalization (VBN). Batch Normalization Layer has been successfully applied in DCGAN [25]. However, the one output's statistics Batch Normalization Layer need is heavily based on other inputs in the same minibatch. It will cause GAN training unstable. To solve this problem, we use Virtual batch normalization proposed by paper [21]. VBN will randomly choose one of the real image minibatch to be the reference batch. The reference minibatch and other minibatch will be normalized by the statistics own to the reference minibatch. The downside of VBN is computationally expensive for twice compute (one forward propagation for reference batch, one forward propagation for current batch) and this trick will only be used in generator.

A. DATASET EXPANDING

We apply Contour Stellar Image as the training dataset. The dataset consists of 4ASK, BPSK, QPSK, OQPSK, 8PSK, 16QAM, 32QAM, 64QAM. Each modulated signal is distributed 10000 images. After 50000 iterations, we got loss of discriminator and generator converged, we get the generated image. Having been given the specific label per row, the generator will give the corresponding image per row. In Fig. 12 (a), (b) the generated image of Contour Stellar Image will be presented, it proves that generator has learned:

We firstly use Contour Stella Image dataset mentioned above to train AlexNet. This result will be the reference. After generating 5000 Contour Stella Image at SNR from -6dB to 14dB per class by ACGAN, we fill them into the

TABLE 1. Test accuracy of baseline and our framework.

SNR(dB)	Original (100%)	Original (50%)	+ACGAN (50%)	Original (60%)	+ACGAN (40%)	Original (70%)	+ACGAN (30%)
0	99.98%	98.17%	**98.18%**	98.50%	98.50%	98.80%	**99.00%**
-2	98.12%	85.27%	**88.05%**	90.49%	**91.46%**	93.84%	**95.00%**
-4	89.52%	73.24%	**77.64%**	76.45%	**80.24%**	82.85%	**85.64%**
-6	82.02%	62.38%	**68.35%**	68.98%	**76.84%**	75.47%	**78.55%**

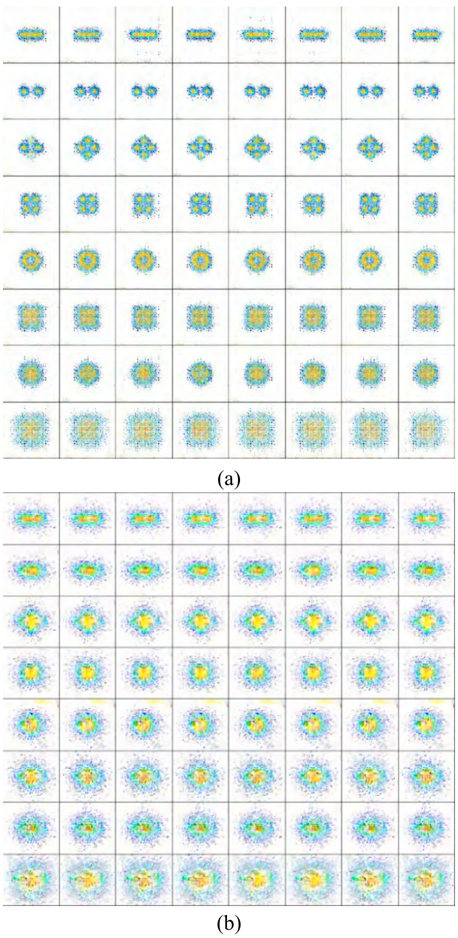

FIGURE 12. Generated Contour Stella Image at different SNR, Each Row Represents One of the class in modulated signal, from top to bottom: 4ASK, BPSK, QPSK, OQPSK, 8PSK, 16QAM, 32QAM, 64QAM. (a) 4dB. (b) −2dB.

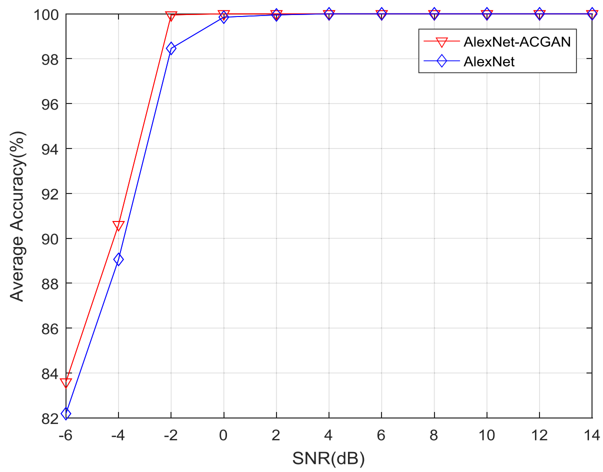

FIGURE 13. The comparison of average accuracy between two frameworks.

original dataset and then feed the new dataset to AlexNet. From Fig. 13, we can have the following observations:

(a) Accuracy has been slightly improved at every SNR
(b) Accuracy Improvement at relatively low SNR is greater than that in relatively high SNR.

The reason of this result is following: (1) there is still improvement space for the baseline result (2) Contour Stella Image is a feature simple image, AlexNet has extracted the most useful feature and additional image generated by ACGAN will just slightly help AlexNet know what is more useful or useless. (3) Contour Stella Image at low SNR are more identical from class to class. So, CNN need more image to extract the feature that counts. That's the reason accuracy improvement at relatively low SNR is greater than that in relatively high SNR.

B. INSUFFICIENT DATASET DATA AUGMENTATION

We also want to explore how the insufficient data influence the output and how generated images improve the result. In this experiment, the baseline result is from AlexNet trained on the original Contour Stella Image mentioned in Section II, which has 10000 real images in dataset. We shrink our dataset by 10%, 20%, 30%, 40% and 50% to figure out how the insufficient data influences CNN. Then we will compensate the insufficient data by using the generated image from ACGAN. The ACGAN will be trained on the remained original images in dataset. It means that dataset components will be: 5000 Original Images+5000 Generated Images, 6000 Original Images+4000 Generated Images, 7000 Original Images+3000 Generated Images, 8000 Original Images+ 2000 Generated Images, 9000 Original Images+1000 Generated Images.

After the ACGAN's training, we feed the AlexNet with dataset mentioned above. Then we got the result as depict in Table (1) and Table (2).

From Table (1) and Table (2), we can observe some fact: (a) the scale of dataset will indeed influence the AlexNet's performance, especially at the low SNR. Due to the fact that

TABLE 2. Test accuracy of baseline and our framework.

SNR(dB)	Original (100%)	Original (80%)	+ACGAN (20%)	Original (90%)	+ACGAN (10%)
0	99.98%	99.79%	**99.80%**	99.80%	**99.82%**
-2	98.12%	96.52%	**97.52%**	98.00%	**98.05%**
-4	89.52%	85.47%	**87.52%**	88.02%	**89.02%**
-6	82.02%	79.01%	**80.25%**	81.55%	**81.85%**

AlexNet is a complex model, it is prone to overfit without sufficient data (b) The data augmentation implement by ACGAN has improves accuracy of classification, especially when the data of different are relatively identical at low SNR and insufficient (c) Contour Stella Image is simple feature images, any noise brought up by ACGAN will have a negative impact on the result, that's the reason AlexNet performance on dataset couldn't surpass the performance on real dataset with the same amount of signal.

VI. CONCLUSIONS AND DISCUSSIONS

In this paper, we explore using ACGAN for data augmentation in task of modulated signals classification task. We propose a new framework for data augmentation by using ACGAN to generate additional images in CNN training. When it comes to the ACGAN training, we utilize several measures to avoid disconverge problem and mode collapse problem. The result of this experiment indicates that our GAN-based data augmentation framework can improve CNN classification and obtain 0.1%$\sim$6% increase in the accuracy.

However, in the future, we will explore other generative model, such restricted Boltzmann machine [26], VAE [27] and other GAN, such CycleGAN [28] to evaluate data augmentation method to apply our model for the general communicate classification.

REFERENCES

[1] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Comput. Netw.*, vol. 50, no. 13, pp. 2127–2159, 2017.

[2] J. Guo, N. Zhao, F. R. Yu, X. Liu, and V. C. M. Leung, "Exploiting adversarial jamming signals for energy harvesting in interference networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 2, pp. 1267–1280, Feb. 2017.

[3] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.

[4] G. Ding *et al.*, "Robust spectrum sensing with crowd sensors," *IEEE Trans. Commun.*, vol. 62, no. 9, pp. 3129–3143, Sep. 2014.

[5] G. J. Mendis, J. Wei, and A. Madanayake, "Deep learning-based automated modulation classification for cognitive radio," in *Proc. IEEE Int. Conf. Commun. Syst. (ICCS)*, Dec. 2016, pp. 1–6.

[6] N. Zhao, F. R. Yu, M. Li, Q. Yan, and V. C. M. Leung, "Physical layer security issues in interference- alignment-based wireless networks," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 162–168, Aug. 2016.

[7] N. Zhao, F. R. Yu, H. Sun, and M. Li, "Adaptive power allocation schemes for spectrum sharing in interference-alignment-based cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 3700–3714, May 2016.

[8] Z. Yang, S. Ping, H. Sun, and A.-H. Aghvami, "CRB-RPL: A receiver-based routing protocol for communications in cognitive radio enabled smart grid," *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 5985–5994, Jul. 2017.

[9] Z. Zheng, T. Huang, H. Zhang, S. Sun, J. Wen, and P. Wang, "Towards a resource migration method in cloud computing based on node failure rule," *J. Intell. Fuzzy Syst.*, vol. 31, no. 5, pp. 2611–2618, 2016.

[10] X. Shi *et al.*, "Graph processing on GPUs: A survey," *ACM Comput. Surv.*, vol. 50, no. 6, 2017, Art. no. 81.

[11] T. Wang, C.-K. Wen, H. Wang, F. Gao, T. Jiang, and S. Jin, "Deep learning for wireless physical layer: Opportunities and challenges," *China Commun.*, vol. 14, no. 11, pp. 92–111, 2017.

[12] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Proc. Int. Conf. Eng. Appl. Neural Netw.*, 2016, pp. 213–226.

[13] S. Peng, H. Jiang, H. Wang, and Y.-D. Yao. *Deep Learning and its Applications in Communications Systems: Modulation Classification*. [Online]. Available: http://personal.stevens.edu/~hwang38/DL.pdf

[14] N. Zhao, F. R. Yu, and V. C. M. Leung, "Opportunistic communications in interference alignment networks with wireless power transfer," *IEEE Wireless Commun.*, vol. 22, no. 1, pp. 88–95, Feb. 2015.

[15] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[16] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 5485–5493.

[17] H. Wu, S. Zheng, J. Zhang, and K. Huang. (Mar 2017). "GP-GAN: Towards realistic high-resolution image blending." [Online]. Available: http://arxiv.org/abs/1703.07195

[18] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 597–613.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 213–226.

[20] X. Zhu, Y. Liu, Z. Qin, and J. Li. (Nov. 2017). "Data augmentation in emotion classification using generative adversarial networks." [Online]. Available: http://arxiv.org/abs/1711.00648

[21] T. Salimans *et al.*, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.

[22] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. (Nov. 2016). "Image-to-image translation with conditional adversarial networks." [Online]. Available: https://arxiv.org/abs/1611.07004

[23] A. Odena, C. Olah, and J. Shlens. (Oct. 2016). "Conditional image synthesis with auxiliary classifier GANs." [Online]. Available: https://arxiv.org/abs/1610.09585

[24] T. White. (Sep. 2016). "Sampling generative networks: Notes on a few effective techniques." [Online]. Available: http://arxiv.org/abs/1609.04468

[25] A. Radford, L. Metz, and S. Chintala. (Nov. 2015). "Unsupervised representation learning with deep convolutional generative adversarial networks." [Online]. Available: https://arxiv.org/abs/1511.06434

[26] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[27] D. P. Kingma and M. Welling. (Dec. 2013). "Auto-encoding variational Bayes." [Online]. Available: https://arxiv.org/abs/1312.6114

[28] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. (Mar. 2017). "Unpaired image-to-image translation using cycle-consistent adversarial networks." [Online]. Available: https://arxiv.org/abs/1703.10593

BIN TANG received the B.S., M.S., and Ph.D. degrees from the College of Computer Science and Technology, Harbin Engineering University, Harbin, China, in 2008, 2011, and 2015, respectively. He holds a post-doctoral position at the College of Shipbuilding Engineering, Harbin Engineering University. His research interests include information theory, security, machine learning, and big data analytics.

YA TU received the B.S. degree from the College of Computer Science and Technology, Taiyuan University of Technology, Taiyuan, China, in 2016. He holds a doctoral position with the College of Information and Communication Engineering, Harbin Engineering University, Harbin, China. His current research interests include signal processing, machine learning, and data analysis.

ZHAOYUE ZHANG received the B.S. degree from the College of Air Traffic Management, Civil Aviation University of China, Tianjin, China, in 2005. He is currently a Lecturer with the College of Air Traffic Management, Civil Aviation University of China. His research interests include signal processing, machine learning, and data analysis.

YUN LIN received the B.S. degree from Dalian Maritime University in 2003, the M.S. degree from the Harbin Institute of Technology in 2005, and the Ph.D. degree from Harbin Engineering University in 2010. From 2014 to 2015, he was a Visiting Scholar with Wright State University, USA. He is currently an Associate Professor with Harbin Engineering University. His current research interests include communication technology, signal processing, information fusion, cognitive radio, and software defined radio.