Using Conditional Adversarial Networks to Deblur the Sonar Image of the Unknown Motion Blur Kernels

Zheping Yan¹, Hang Zhao*¹ and Wenyue Hu¹

¹College of Automation, Harbin Engineering University, Harbin, Heilongjiang Province, 150001, China
*Correspondence: hangzhao@hrbeu.edu.cn

Abstract. In order to recover the blur sonar image collected by the side scan sonar during motion, we propose a solution based on the conditional adversarial networks to deblur the sonar image of the unknown motion blur kernels. First, we use improved conditional adversarial networks to recover the sonar image, and improve the loss function, so that the quality of image generation is improved while the training stability is enhanced. Then we propose a method for generating blurred sonar images. The blurred sonar image generated by this method is closer to the real blurred sonar image. Finally, we made our own sonar image set and trained it with two-timescale update rule. The final results proved that the image restored by this method has higher definition.

1. Introduction

The side scan sonar[1] during motion may blur the sonar image of underwater moving objects. If the movement speed is too fast, the collected sonar image will be too blurred to extract the required information. In addition, although the sonar image is substantially identical to the optical image, it is a plane or spatial distribution of energy. Due to the complex or variable nature of the seawater medium of the acoustic information transmission channel boundary[2-3], and the projection characteristics of the sonar itself, the energy distribution of the sonar image is irregular. Generally, the sonar image is a low frequency image, and the details of the image are not much, which makes the information in the sonar image generally difficult to recognize. The sonic image deblurring can improve the accuracy of the recognition of the marine environment, so that the collected sonar image information can be fully utilized.

The blurred sonar image can be expressed as the sum of the convolution and additive noise of the unknown blur kernels and the sharp sonar image. Its mathematical model expression is equation (1):

\[ X_b = \text{Kernel}(M) \ast X_c + \text{Noise}. \]  

(1)

Where \( X_b \) is a blurred sonar image, \( \text{Kernel}(M) \) are unknown blur kernels determined by the motion field \( M \), \( X_c \) is the sharp sonar image, \( \ast \) represents a convolution operation, and Noise represents an additive noise.

Previous image deblurring algorithms use local linear assumptions of fuzzy functions and simple heuristics to quickly estimate unknown blur kernels. For example, Sun et al.[4] use convolutional neural networks(CNN) to estimate the blur kernels. Chakrabarti[5] predicts that the complex Fourier coefficients of the motion kernels performed non-blind deblurring in Fourier space. And Gong[6] uses a full convolutional network to move motion flow estimates. All of these methods use CNN to estimate a particular unknown fuzzy function.
With the development and growth of deep learning, Goodfellow et al. [7] proposed a framework for image style migration, called the Generative Adversarial Networks (GAN), which divides the deep neural network into two parts: the generation network G and the discrimination network D. G and D form a competing network: G receives random noise for generating pseudo samples. D receives the real samples and the generated samples and tries to distinguish them. In the process of competing two networks, the capabilities of the two networks are getting stronger and stronger: the pictures generated by the G network are more and more like real pictures, and the D network is more and more easy to distinguish the true and false images. The mathematical model of GAN is equation (2):

$$\min_G \max_D \mathbb{E}_{x \sim P_r}[\log(D(x))] + \mathbb{E}_{z \sim P_z}[1 - \log(D(G(z)))].$$  \hspace{1cm} (2)

Where $P_r$ is the data distribution, $P_g$ is the model distribution, $\hat{x} = G(z)$, $z$ is a sample of simple noise distribution.

Today the application of GAN is no longer limited to conversion between two images with different objects. More and more improvements are now being made to control the output by controlling the user's input. For example, Mirza et al. [8] propose a conditional confrontation network, which is an extension of the original GAN, with additional information $y$ added as conditions to the original generator and discriminator to implement the condition GAN. Based on the conditional confrontation network, Lsola et al. propose the pix2pix [9] structure and the pix2pixHD [10] structure, which use the U-Net [11] architecture to generate high-definition images. Zhu et al. [12] complete the transformation of image style through conditional confrontation network without pairing training data.

2. Related algorithms in this paper

In our paper, the generated network structure $G$ is used to map motion blur sonar images to sharpened sonar images. The discrimination network $D$ is used to distinguish the generated sonar image from the real high definition sonar image. Alternately train $G$ and $D$ separately to minimize the loss function of $G$ and $D$, so that the high-resolution sonar image generated by $G$ is confused with the judgment of $D$ as much as possible and $D$ judges as much as possible. When the loss function values of $G$ and $D$ are almost constant, the training is stopped.

2.1. Network structure

We input the blurred image into $G$ to obtain the restored image, then input the image we generated together with our desired image into $D$ to obtain the corresponding loss, and finally feedback the loss back to the $G$ network and the $D$ network.

The network structure is shown in Figure 1.

![Network Structure](image)

Figure 1. The network structure used in this article.

$G$ consists of three main parts: a multi-scale block part, a feature extraction block, and an upsampled block. In the multi-scale block section, the data set is enhanced by a series of random cropping and scaling operations, effectively avoiding network overfitting. In the feature extraction block part, the feature information of the motion blur sonar image is extracted by downsampling the three convolutional blocks and a series of residual blocks. The upsampled block is upsampled by a series of deconvolution blocks to achieve feature fusion and the feature is restored to the same size as the original image. Image-level features are fused at the end of $G$ to overcome the problem of effective
weight reduction over long distances. And batch normalization (BN)[13] is added in G, which solves the problem that the gradient disappears or explodes due to the change of the data distribution of the middle layer in the training process. Dropout[14] is added after each convolution block to prevent model overfitting. The G network structure is shown in Figure 2.

![Figure 2. Generating a network G structure.](image)

The residual block is the residual block in ResNet[15]. Except that the first convolutional convolution kernel and the last convolutional convolution kernel are 7 × 7, the remaining convolution kernel (or deconvolution kernel) is 3 × 3. The number of convolution kernels (or deconvolution kernels) and the convolution step size of each convolution are: (64, 1) → (128, 2) → (256, 2) → (256, 1) → (256, 1) → (256, 1) → (256, 1) → (256, 1) → (256, 1) → (256, 1) → (128, 2) → (64, 2).

D only deals with the classification of generated sonar images and real sonar images, so we use a simple two-class model. D is composed of a plurality of consecutive convolution blocks and one fully connected block. BN is added to D. Dropout is added after each convolution block to prevent model overfitting. The fully connected layer uses ReLU and sigmoid as the active layer, and the leaky block uses LeakyReLU as the active layer. D network structure is shown in Figure 3.

![Figure 3. Discriminating network D structure.](image)

All convolution kernel sizes are 4 x 4. The number of convolution kernels and the convolution step are in order: (64, 2) → (128, 2) → (256, 2) → (512, 1) → (512, 1). The output level of the full connection layer is 1024→1.

2.2. Loss function

Traditional confrontational losses focus on restoring picture texture details without considering the connection between pixels. Most researchers[16-18] use regular loss to guarantee the commonality of input and output, but the characteristics of regular loss lead to the loss of a part of the local details while improving the consistency of the global content. The regular loss mathematical expression is equation (3):

$$ L(G) = \frac{1}{2m} \sum_{i=1}^{m} || I_{i} - I_{i} ||_{k}. \quad (3) $$

Where $I_{i}'$ is the generated image, $I_{i}$ is the real image, m is the number of generated images, k=1 or 2, k=1 indicates that L(G) is the L1 regular loss, and k=2 indicates that L(G) is the L2 regular loss.

In our paper, the loss function of G should consider both the local reconstruction accuracy of each pixel and the global pixel combination position state, that is, the relationship between each pixel and other global pixels. Therefore, this paper introduces the global information loss function $loss_{content}$.
while using the $\text{loss}_{\text{content}}$ [19] guarantee local details, which is an improved L2 regularization loss based on the difference between the generated image features and the target image features, so that the picture achieves better vision. The effect of $\text{loss}_{\text{content}}$ not only considers the global content loss, and also takes into account the local detail loss. The mathematical expression of $\text{loss}_{\text{content}}$ is equation (4):

$$
\text{loss}_{\text{content}} = \frac{1}{W_{ij}H_{ij}} \sum_{x=1}^{W_{ij}} \sum_{y=1}^{H_{ij}} (\Phi_{ij}(l)_{x,y} - \Phi_{ij}(l)_{x,y})^2.
$$

(4)

Where $\Phi_{ij}(l)$ indicates that the target image obtains the feature result through VGG[20].

Mao et al.[21] believe that the traditional GAN loss function does not make the collected data distribution close to the real data distribution. They propose to use the least squares loss instead of the cross entropy loss in GAN to solve the two defects of the standard GAN generated image quality and the instability of the training process. In view of the outstanding advantages of the least squares loss, we replace the cross entropy loss in the loss with the least squares loss. The mathematical expression against the loss function $\text{loss}_{\text{adversarial}}$ is equation (5):

$$
\text{loss}_{\text{adversarial}} = \frac{1}{2} E_{x \sim P_l} [D(G(x)) - a]^2.
$$

(5)

Finally, we combine $\text{loss}_{\text{adversarial}}$ and $\text{loss}_{\text{content}}$ according to a certain ratio. The mathematical expression of the loss function of $G$ in this paper is equation (6):

$$
L_G = \alpha \text{loss}_{\text{adversarial}} + \beta \text{loss}_{\text{content}}.
$$

(6)

Similarly, we also consider the stability of training when designing the loss function $L_D$ of the discriminant network. We also used the least squares loss instead of the cross entropy loss. At the same time, we refer to the idea put forward by Gulrajani et al.[22], adding Wasserstein distance loss with gradient penalty in $L_D$ to prevent gradient explosion and stable training. The mathematical expression of the loss function of network $D$ is equation (7):

$$
L_D = \mu \left( \frac{1}{2} E_{x \sim P_l} [D(x) - b]^2 + \frac{1}{2} E_{x \sim P_l} [D(G(x)) - c]^2 \right) + \gamma E_{x \sim P_{ij}} [||\nabla_x D(x)||_p - 1]^2.
$$

(7)

2.3. Sonar image data pair set production

A simple way to obtain a data pair is to use the side scan sonar to capture a sharp sonar image and a blurred sonar image in a very short time at the same location. It is unrealistic to obtain two images in the same place due to the influence of the water flow and the movement of the submarine itself. So the method used to create a dataset in our paper is to generate the blurred sonar images dataset on a sharp sonar images dataset.

Because the points in the blurred image are related to the corresponding points in the original sharp image and the points in the moving direction of the sharp image, and the closer to the anchor point, the greater the point influence in the direction of motion, we consider the position of the anchor while designing the fuzzy kernel. In addition, due to the interference of the underwater environment, the blurred sonar image carries additive noise, so we add the simulated underwater environment interference noise when generating the blurred image.

The pseudo code of the algorithm for generating fuzzy sonar images in our paper is as follows:

**Algorithm 1 Create blur image**

Parameters:
Velocity - Side scan sonar relative speed
Angle - Side scan sonar relative motion angle
EPS - Very small interference holder - Correction threshold

1: **procedure** Create_blur(Image, Velocity, Angle, holder)
2:   sx = [ | Velocity * cos(Angle) + a * holder - Velocity * EPS | ]
3:   sy = [ | Velocity * sin(Angle) + a * holder - Velocity * EPS | ]
3. Experiment and analysis

In the past GAN network training, each iteration needs to update D several times before updating G, so that the gradient does not disappear when training G. In other words, we update the discriminator several times before updating the generator, for example, the regularization discriminator may require 5 or more updates before each generator update. And adding regularization loss to the discriminant network will make GAN training slower. For the problem of slow learning and unbalanced update steps, Zhang et al.[23] propose different learning rates for D and G. Training D with a higher learning rate, because a higher learning rate can solve the problem of slow learning by the regularization discriminator. G is trained using a lower learning rate because a lower learning rate can improve the accuracy of image generation. The two-timescale update rule (TTUR) used by Zhang et al. allows G and D to be updated at 1:1 speed. Given the advantages of TTUR, we use different learning rates to optimize our discriminator and generator.

We have accumulated relevant sharp sonar images through previous experiments, and made our own sonar images through the blurred image generation algorithm. By testing different sonar images, we got very good results. In Figure 4, we present two sets of comparative experimental pictures. Each group consists of a blurred image, a restored image and a clear image. In each group, we highlight the contrast of three small areas.
The results show that the proposed scheme can deblur the motion blur sonar image, and the effect of this model is still very good under the sonar image with different degrees of blur.

We present the values of the average peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) based on our data set. The results are shown in Table 1.

Table 1. Comparison of the results of training different models on our datasets.

|        | Sun et al. | Nah et al. | Our result |
|--------|------------|------------|------------|
| PSNR   | 23.4       | 26.9       | 27.2       |
| SSIM   | 0.817      | 0.858      | 0.916      |

We compared the model structure proposed by Sun et al.[24], the model structure proposed by Nah et al.[25], and the model structure we used. Through comparative analysis, our results performed very well in terms of structured self-similarity and achieved good results in peak signal-to-noise ratio. It is slightly better than some of the existing models.

4. Conclusion
We proposed a solution for generating a sharp sonar image based on conditional adversarial networks. And for how to make an effective blur-sharp sonar data set, we described an effective method and gave the corresponding pseudo code. For the effectiveness of training, we used a dual learning rate approach to train our model. The final experimental results are in line with the expected results.

Acknowledgments
We gratefully acknowledge assistance and encouragement from teachers and classmates. We also want to thank the teachers and students for their comments and suggestions. These valuable suggestions have enabled us to complete this paper better.
References
[1] Barngrover, C., Althoff, A., Deguzman, P., Kastner, R. (2016). A brain–computer interface (bci) for the detection of mine-like objects in sidescan sonar imagery. IEEE Journal of Oceanic Engineering, 41(1), 123-138.
[2] Mason, S., Berger, C. R., Zhou, S., Willett, P. (2008). Detection, synchronization, and doppler scale estimation with multicarrier waveforms in underwater acoustic communication. IEEE Journal on Selected Areas in Communications, 26(9), 1638-1649.
[3] Singer, A. C., Nelson, J. K., Kozat, S. S. (2009). Signal processing for underwater acoustic communications. Communications Magazine IEEE, 47(1), 90-96.
[4] Sun, J., Cao, W., Xu, Z., Ponce, J. (2015). Learning a convolutional neural network for non-uniform motion blur removal. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, pp. 769-777.
[5] Chakrabarti, A. (2016). A neural approach to blind motion deblurring. In: Computer Vision – ECCV 2016. Amsterdam, pp. 221-235
[6] Gong, D., Yang, J., Liu, L., Zhang, Y., Reid, I., Shen, C. (2017). From Motion Blur to Motion Flow: A Deep Learning Solution for Removing Heterogeneous Motion Blur. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu. pp. 3806-3815.
[7] Goodfellow I J, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. (2014). Generative Adversarial Networks. In: Neural Information Processing Systems(NIPS), Montreal. pp. 2672-2680.
[8] Mirza, M, Osindero, S. (2014). Conditional Generative Adversarial Nets. https://arxiv.org/abs 1411.1784.
[9] Isola, P., Zhu, J. Y., Zhou, T., Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu. pp. 1125-1134.
[10] Wang, T. C., Liu, M. Y., Zhu, J. Y., Tao, A., Kautz, J., Catanzaro, B. (2018). High-resolution image synthesis and semantic manipulation with conditional gans. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Salt Lake City. pp. 8798-8807.
[11] Ronneberger O., Fischer P., Brox T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Munich, pp. 234-241.
[12] Zhu, J., Park, J., Isola, P., Alexei A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu. pp. 2223-2232.
[13] Ioffe, S., & Szegedy, C. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. https://arxiv.org/abs/1502.03167.
[14] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929-1958.
[15] He K., Zhang X., Ren, J., Sun, J. (2016). Deep Residual Learning for Image Recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). City of Las Vegas. pp. 770-778.
[16] Li, C., Xu, K., Zhu, J., Zhang, B. (2017). Triple generative adversarial nets. http://papers.nips.cc/paper/6997-triple-generative-adversarial-nets.
[17] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X. (2016). Improved techniques for training gans. http://papers.nips.cc/paper/6124-improved-techniques-for-training-gans.
[18] 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: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Salt Lake City. pp. 8789-8797.
[19] Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu. pp. 4681-4690.

[20] Simonyan, K., Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. https://arxiv.org/abs/1409.1556.

[21] Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., Smolley, S. P. (2017). Least squares generative adversarial networks. In: The IEEE International Conference on Computer Vision (ICCV). Venezsia. pp. 2794-2802.

[22] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A. (2017). Improved training of wasserstein gans. http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.

[23] Zhang, H., Goodfellow, I., Metaxas, D., Odena, A. (2018). Self-attention generative adversarial networks. https://arxiv.org/abs/1805.08318.

[24] Sun, J., Cao, W., Xu, Z., & Ponce, J. (2015). Learning a convolutional neural network for non-uniform motion blur removal. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston. pp. 769-777.

[25] Nah, S., Kim, T. H., Lee, K. M. (2017). Deep multi-scale convolutional neural network for dynamic scene deblurring. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu. pp. 3883-3891.