2nd International Conference on Industrial Artificial Intelligence (IAI), Oct. 23-25, 2020, Shenyang, China

Pre-processing for UAV Based Wildfire Detection: A Loss U-net Enhanced GAN for Image Restoration

Linhan Qiao[1], Youmin Zhang[2*], Yaohong Qu[3]
[1]*Department of Mechanical, Industrial and Aerospace Engineering, Concordia University*, Montreal, Canada
742954173@qq.com
[2]*Department of Mechanical, Industrial and Aerospace Engineering, Concordia University*, Montreal, Canada
ymzhang@encs.concordia.ca*
[3]*School of Automation, Northwestern Polytechnical University*, Xi'an, China
qyh0809@126.com

***Abstract*—In this paper, a U-net with feature loss enhanced generative adversarial network (GAN) is designed for the wildfire or smoke images restoration which is captured by unmanned aerial vehicles in a serious environment. Based on the concepts of GAN, feature loss, and fastai API, we firstly crappy the target images, and train a U-net architecture based generator, then load the adaptive loss of discriminator and the mean square error together to train the GAN model. After the GAN, a second U-net grabs the feature loss from an Imagenet pre-trained loss network to generate the GAN output images with one more step. This U-net enhanced the generator of GAN and helped to get the main features in human conception. Comparing with other restoration methods, this model used the adaptive loss to train the GAN and perceptual loss to train the next U-net. Learning rate with simulation annealing helped jumping out of the local minimum. The result proved the good performance of this model.**

***Index Terms*—GAN, spectral normalization, feature loss.**

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) are usually being used in wildfire detection because of their flexibility and resources saving [1]. However, because of the complexity of the forest environment and some disturbances and noises, like winds or UAVs' vibrations, the captured images or videos may be crapped, these effects may finally make the low quality or blurred images, and will affect the assessment to wildfire damages and may lead to some economical even human life's losses. Actually, in 1990s, it is already concluded by Kulkarni that the recorded images are always distorted or of finite resolution. An inversion of the kernel representing the system, in the presence of noises, is an ill posed problem. The direct inversion often yields an unacceptable solution [2]. Of course, more costs on sensors or hardware can help to deal with this problem. But sometimes, it may be not acceptable to have more hardware equipped with UAV due to the size, weight, and cost limitations etc. in these kinds of situations, efficient and effective algorithms are needed to help for increasing the quality of the images or videos captured.

Traditionally, there are many methods [17] [19] to deal with the image degradation process in terms of a matrix-vector equation (1), which appears in the work of Banhams et al. [3].

$$y = Hf + n \tag{1}$$

where y, f and n are the observed, original, and noisy images. It is assumed that the original image is of support $N \times N$, the vectors then have support $N^2 \times 1$, and H represents the $N^2 \times N^2$. Paper [3] noted that classical direct approaches for solving equation (1) is to find an estimate $\hat{f}$ which minimizes the norm

$$\|y - H\hat{f}\| \tag{2}$$

This thought leads directly to the generalized inverse filter [18], which is given by the solution to

$$(H^T H)\hat{f} = H^T y \tag{3}$$

A significant amounts of research devoted to Gaussian denoising, which corresponds to (1) with $H = I$ [5]. These methods efficiently worked on the influence of noises. For other situations, like warp, lighting and so on, external data sets or data augmentations have been used to train a network [20] [21] [22]. These works also achieved a lot, especially with the emergence of deep neural networks (DNNs). For example, the very deep models such as enhanced deep residual networks for single image super resolution (EDSR) in [6]–[9] which are being widely studying.

However, it is noted that the need of deeper network means more computational resource and it is observed that the smooth region with mild noise could be well restored with a 5-layer CNN [10]. How to restore a blurred image and get the main features of target with a smaller network is still a very challenging problem.

Focusing on this problem, we desire a new kind of loss function. In this paper, a generative adversarial network (GAN) [12] based generator discriminator with adaptive loss is considered because the GAN model works well to the motion-blurred images [22]. Then another U-net with perceptual loss is considered to enhance the GAN capture main features. For the GAN, a pre-trained U-net is firstly designed as a generator. Then, a binary classification model is designed as a discriminator, which learns to classify which is the high quality image and which is the generated image during training process so that it could help to fine tune the generator more. Instead of the pixel mean square error (MSE) between the high quality image and the generated image, the loss changed into

978-1-7281-8216-2/20/$31.00 ©2020 Crown

something that describes the difficulty level of separating these images of the discriminator. As the training goes into more steps, generated images will be "shown" to the discriminator and get fixed again and again. To ensure the generated images contain main features of the target images, another U-net is trained with feature loss, which are the activation one step before the max pooling in a loss network. They are suspected containing feature maps one step before their changing. This model first used a kind of adaptive loss GAN transformed the problem into a classifier recognition confusion task. And the feature loss guaranteed the preservation of the main features of factual wildfire or smoke. The result proved its performance.

The arrangement of this paper is organized as follows. In Section 2, the model of this generative adversarial network (GAN) is discussed. In Section 3, the training and testing process are shown, Section 4 is the conclusion.

II. The U-net GAN Loss Model

The model can be separated into 3 parts: generator, discriminator and a loss U-net. The generator is designed as a Resnet architecture U-net, with mean square error (MSE) loss of pixel values between the high quality images and generated images. The discriminator is a binary softmax classifier, with adaptive cross entropy loss, which will try to divide the generated image and original target image after every step of training. The loss U-net is a U-net with weight combination loss behind the GAN ensures the whole model to capture mattering features of foreground objects and backgrounds. The model structure can be briefly described by Fig. 1.

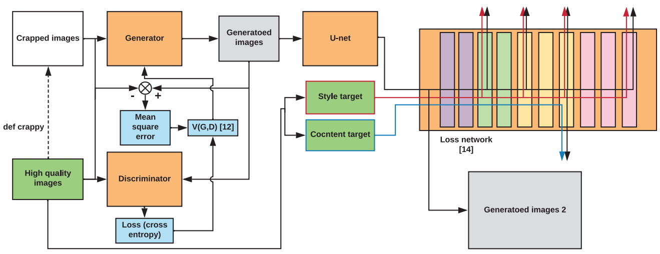

Fig. 1. Brief structure of loss U-net GAN based image restorer. A loss U-net enhance GAN model to understand more features of target images.

In this work, the training data set consists of high quality wildfire smoke images and their crapped copies.

The generator (G) of this generative adversarial network (GAN) is a pre-trained U-net based on Resnet34 architecture, because Resnet have a standard encoder structure, which is appropriate as the encoding part or down sampling part of U-net.The reason we choose U-net as a generator is that U-net always perform good in segmentation which is to generate a new foreground mask, just like our restoration tasks, generate some missing part of the target image. Training of G is paused after one epoch when it is trained to get a generated image and mean square error (MSE) between the high quality image or our target. MSE is to compare the pixel values of generated image and high quality image, it can be written in a form of equation (4).

$$\mathrm{MSE}(y,\hat{y}) = \frac{\sum_{i=0}^{N}(y_i-\hat{y}_i)^2}{N} \tag{4}$$

The generated image and target are then given to discriminator (D), where D is a binary classifier. If D can easily recognize the generated image, where the binary cross entropy loss is very low, it means that the generated image is not good enough. In another word, loss of G will be higher as loss of D is lower, loss of G will get lower when the loss of D is higher. So, the generative adversarial network (GAN) is to balance these two losses.

Value function $V(G,D)$ can be used to balance the loss of generator (G) and the discriminator (D) to describe the loss or cost of GAN. It is proposed in paper [12] and now being widely used in GAN model building.

$$\min_G \max_D V(G,D) = \mathbf{E}_{x\ p_{data}x}[\log D(x)]+ \\ \mathbf{E}_{z\ p_z z}[log(1-D(G(z)))] \tag{5}$$

The explanation of this equation is stated as follow: To learn the generator's distribution p_G over data x, a prior on noise variables $p_z(z)$ is defined, then represent a mapping to data space as $G(z;\theta_G)$, where G is a represents the generator with parameters θ_G. In the same way, discriminator $D(x;\theta_D)$ is defined. $D(x)$ represents the probability that x is loaded from the data rather than p_G. D is trained to maximize the probability of assigning the correct label to both training examples from G. G is trained to minimize $\log(1-D(G(z)))$.

GAN transfers the pixel comparing problem of generator into a problem which is to generate images to confuse the classifier or discriminator. The discriminator is designed as a standard classification model with the help of fastai. After saving the generated images from generator, normalize these images and target images into data bunches. Binary cross entropy (BCE) loss [13] is used for this classifier

$$\mathrm{BCE}(y,p) = \begin{cases} -\log(p) & \text{if } y=1 \\ -\log(1-p) & \text{otherwise} \end{cases} \tag{6}$$

where $y \in \pm 1$ specifies the ground truth class, $p \in [0,1]$ is the probability of the estimation of the discriminator model from the label $y=1$.

After building generator and discriminator, what we need to do is to connect them together or switch between them. In a generative adversarial network (GAN), the classifier can not just be a pre-trained soft-max, because the generator and the discriminator can not increase the weights in the same direction, or it will make the GAN finally out of control. How to keep the GAN work stable is a problem. Spectral normalization can be used to keep the GAN training stability [14], and in the main time the discriminator need to deal with a kind of adaptive loss.

Assume the discriminator can be written in the form of equation (7)

$$f(x,\theta) = W^{L+1} a_L(W^L(a_{L-1}(W^{L-1}(\cdots a_1(W^1 x)\cdots)))) \tag{7}$$

where x represents the input, $\theta = \{W^1, W^2 \cdots W^L, W^{L+1}\}$ is the learning parameter, a_L is the activation of L th layer. To make unity of form of equation (5), the final output of discriminator could be

$$D(x,\theta) = \mathcal{A} f(x,\theta) \tag{8}$$

For the fixed generator G, the optimal discriminator for this form is given by

$$D_G^*(x) = \frac{q_{data(x)}}{q_{data}(x) + p_G(x)} = \text{sigmoid}(f^*(x)) \tag{9}$$

where $f^*(x) = \log q_{data}(x) - \log p_G(x)$, q_{data} is the data distribution, p_G is the generator distribution. The derivative of $f^*(x)$ is

$$\nabla_x f^*(x) = \frac{1}{q_{data}(x)} \nabla_x q_{data}(x) - \frac{1}{p_G(x)} \nabla_x p_G(x) \tag{10}$$

This derivative is unbounded or even not compute-able, Lipschiz constant of discriminator can be used as a regularity condition [14]. Spectral normalization controls the Lipschiz constant by literally constraining the spectral norm of each layer. For the discriminator (D) from the set of K-Lipschiz continuous function, what we want for equation (5) is

$$\underset{\|f\|_{Lip} \leq K}{\arg} \max V(G,D) \tag{11}$$

Use the inequality $\|g_1 \circ g_2\|_{Lip} \leq \|g_1\|_{Lip} \cdot \|g_2\|_{Lip}$ to observe the bound on $\|f\|_{Lip}$

$$\begin{aligned} \|f\|_{Lip} \leq & \|h_L \mapsto W^{L+1} h_L\|_{Lip} \cdot \|a_L\|_{Lip} \\ & \cdot \|h_{L-1} \mapsto W^L h_{L-1}\|_{Lip} \\ & \cdots \|a_1\|_{Lip} \cdot \|h_0 \mapsto W^1 h_0\|_{Lip} \\ = & \prod_{l=1}^{L=1} \|h_{l-1} \mapsto W^l h_{l-1}\|_{Lip} \\ = & \prod_{l-1}^{L=1} \sigma(W^l) \end{aligned} \tag{12}$$

where $h_{in} \mapsto h_{out}$ is the spectral norm of each layer, $\sigma(W^l)$ is the spectral norm of weight matrix W of layer l. Spectral normalization normalizes the spectral norm of W so that it satisfies the Lipschiz constraint $\sigma(W) = 1$.

After discriminator critic, the generator train the model again, and get a new generated image. If the binary cross entropy (BCE) loss increased, it means the generated image looks good enough that the discriminator can not obviously tell the difference between these two. If the discriminator loss get lower, the generator will work harder. If the generator work harder, the critic will be harder, the loss will increase again. So, we check the output of GAN every epoch until it is acceptable.

When the generator and the discriminator is pre-trained, they can be hooked together to train the whole GAN model. Based on the concept of equation (5), and to avoid the generated images have no relations to the target images, the loss function used for this GAN learner is the sum of discriminator BCE loss and pixel MSE loss multiplied a number. To keep these two in same scale, the number is experimentally from 5 to 200. After building the architecture of GAN, fit function in fastai can help us to find appropriate learning rates. Then we need to solve the problem that how the generated images contain the mattering features we concerned of these target images.

Paper [15] put the prediction images from a transform network and target images through an Image-Net pre-trained loss network, VGG16, and grab the activation in the middle of loss network, where these activation might be feature maps. Johnson, et al. proposed a perceptual loss in that paper to hook these two convolutional neural networks (CNNs). So we use the feature loss which is from the pre-trained VGG model to train a Resnet architecture U-net to get final generated images which to some extend understands the features of target images.

Based on the perceptual loss concept in paper [15], we build the third part, a loss U-net, the loss network Φ to train U-net is used to define loss functions $l_1 \cdots l_k$. Assume the U-net is parameterized by weights W, and generates the output $\hat{y}_{middle}$ of GAN into final output $haty$ with $\hat{y} = f_W(\hat{y}_{middle})$. Every loss computes a scalar value $l_i(\hat{y}, y_i)$, then the U-net training process can be seen to minimize a weighted combination loss through stochastic gradient descent:

$$W^* = \arg\min_W E_{\hat{y}_{middle}, y_i}[\sum_{i=1} \lambda_i l_i(f_W(\hat{y}_{middle}), y_i)] \tag{13}$$

The loss network Φ defines feature reconstruction loss l_{feat}^{Φ} and style reconstruction loss l_{style}^{Φ} [15]. In this paper we do not discuss about the style feature, so l_{style}^{Φ} is ignored. Rather than encouraging the pixels, this U-net encouraging the middle generated images to have similar feature representations by Φ. Assume j is a convolutional layer, $\Phi_j(\hat{y}_{middle})$ will be a feature map in shape of $C_j \times H_j \times W_j$, the feature reconstruction loss is the squared, normalized Euclidean distance between feature representation:

$$l_{feat}^{\Phi,j}(\hat{y}, y) = \frac{1}{C_j H_j W_j} \|\Phi_j(\hat{y}) - \Phi_j(y)\|_2^2 \tag{14}$$

III. Training and Testing

The experiment of this paper is worked on Google compute platform with fastai API. Similar with Pytorch and Tensorflow, fastai is a deep learning library which provides practitioners with high-level components that can provide state-of-the-art results in standard deep learning domains, it can be mixed and matched to build new approaches [16].

The reason why we choose fastai is that it firstly used some new algorithms and tricks that can straightly being applied with NVIDIA CUDA, like the simulated annealing learning rate, the hook between loss network and U-net in this paper and half position compute.

The data sets we are using are downloaded from Google Image, after downloading by sets with a Java trick, we sort out most of high quality forest smoke or fire images as our target images. Then, these high-quality images are crapped with a crappy function. Flip, lighting and other kinds of transform are added as a data augmentation. Fig. 2 is the result of our crapping, the pixels are decreased from 256×256 to 96×96 and a number watermark from 10 to 80 is randomly drawn on the random location of the image to simulate some data lost or disturbance situation. If the number is smaller, it means the pixel decreased more. From Fig. 2, we can see that the images are crapped. The next step is to increase the crapped images' quality and remove the number watermarks on them.

Fig. 2. Crappy function works on wildfire images, by editing different parts of parameters to simulate the UVA captured low quality images.

To pre-train the generator, we build the U-net based on a Resnet34 architecture. Pre-training helps the model to understand the rough features of the objects or material and slightly accelerated the training. The loss function of this U-net is mean square error (MSE) loss. Regularization for this U-net is weight decay [24], because we are going to build a generative adversarial network (GAN), in which the discriminator of GAN with adaptive gradient, it might lead to worse generalization than stochastic gradient decent (SGD) with momentum, however it is proved that the GAN can not work well with momentum [14], which is similar to the Adam in paper [24], that Adam with decoupled weight decay yields substantially better generalization performance. The decay value we set $1e^{-3}$ because Resnet model is proven that it usually works well with learning rate from $4e^{-4}$ to $3e^{-3}$. After 7 epochs training, the validation mean square error (MSE) loss reached 0.058, it is shown in Table.I. Then we unfreeze the network, which means to use the whole U-net but not only encoder part to learn. Fig. 3 showed the entire U-net loss changing. In this figure, the left is of frozen encoder part of U-net and the right is of the unfreeze encoder-decoder U-net. There are two points we need to focus on the right

TABLE I
PRE-TRAINED ENCODER PART OF GENERATOR TRAINING LOSS AND VALIDATION LOSS

epoch	train loss	valid loss	time
0	0.673526	0.135339	00:51
1	0.322672	0.083597	00:23
2	0.198912	0.070252	00:24
3	0.139082	0.064365	00:23
4	0.107195	0.063124	00:23
5	0.088047	0.059863	00:23
6	0.075607	0.058513	00:23

chart. By unfreezing, opening up sampling parts of U-net, after the training goes 6 more step annealing slide the learning rate from $1e^{-6}$ to $1e^{-3}$, the validation loss jumped out of local minimum points for times and finally got somewhere around 0.054, training loss reached 0.052. However, in some where of this process, the validation losses are bigger than training losses, but it is hard to know whether it means over trained or our model is well fit, because it surely jumped out of local minimum. How to judge this performance need more discussion.

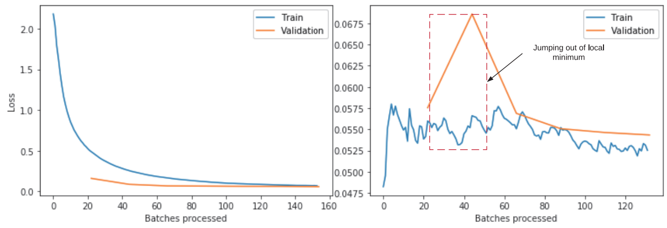

Fig. 3. U-net training loss and validation loss changing. Left: Freeze encoder part of U-net; Right: Unfreezed encoder-decoder U-net. It seems over trained but jumped out of local minimum.

Fig. 4 shows the only generator output. We can see that the generator successfully removed the number water marks on the crapped images, but the pixels are still low, images look blur. Nonrigid bodies like smoke or fire do not have obvious edges, the generator is hard to grab their main features and even affect our judgement of forest or backgrounds. We hope the discriminator can help us to improve these low quality images more.

After building a working generator, we design the discriminator by "gan-critic", which is a function of fastai to find a GAN suitable binary classifier. The inputs of this discriminator are the output of generator and target images, its mission is to classify these two kinds of images. We give the batch size little smaller to avoid running out of GPU ram. Fig. 5 is one data batch we input to train the discriminator. These images are labeled with two classes,"tagets" or "image gen" and separated into training set and validation set by 0.1.

Build the discriminator with learner as usual, but the loss need to be adaptive loss, not just binary cross entropy loss. This loss will expand the target to match the output size before applying the discriminator. Still, we use a weight decay regularization to avoid over fitting, after all these things are

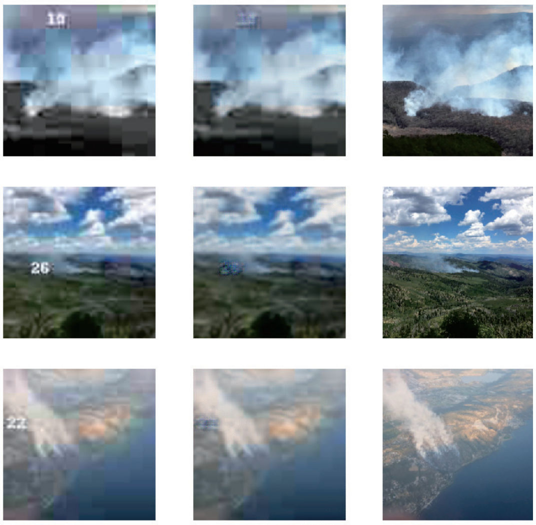

Fig. 4. Generator output. Crapped images, generated predictions, and targets. The watermarks are nearly removed, but the generated images are still in low pixel by comparing with the target.

Fig. 5. One batch to train discriminator. The batch size is set 9, smaller than the batch size to generator 16. The batch size can be bigger with more powerful GPU.

done, we can train this critic by iterative. Through a trick of fastai [16], we find an appropriate learning rate for the discriminator through Fig 6 to be $4e^{-4}$.

After 5 epochs' training, the discriminator can recognize targets and predictions pretty well. Table. II shows its loss and accuracy in each epoch.

Combine the generator and the discriminator we get a generative adversarial network (GAN) model that can restore the crapped wildfire images. When the training of GAN is finished, the testing find that the number watermark is removed and the blurred images slightly increased pixels. However, as

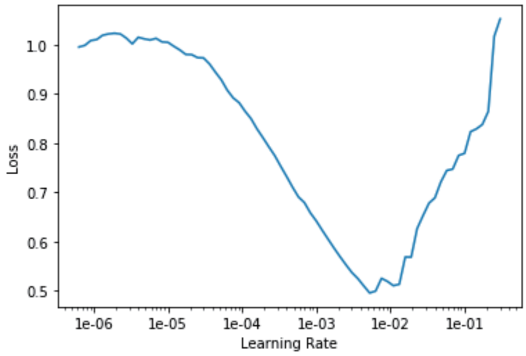

Fig. 6. Loss changing with different learning rate. By finding the lowest point $*10^{-1}$, get the appropriate learning rate could be $4e^{-4}$.

TABLE II
DISCRIMINATOR TRAINING LOSS, VALIDATION LOSS, AND ACCURACY

0 epoch	train loss	valid loss	accuracy thresh expand	time
0	1.050675	1.153025	0.533333	00:04
1	0.867651	0.563016	0.757576	00:03
2	0.671420	0.157411	0.933333	00:03
3	0.551113	0.046685	0.987879	00:03
4	0.452396	0.044083	0.987879	00:03

Fig. 7 shows, the output images look a little weird in human cognition. Because the output images or predictions just find some way to make the images sharped, the generator can not catch the mattering features in target images in human cognition. This is the reason why we need a feature loss or perceptual loss network enhanced U-net to grab more main features.

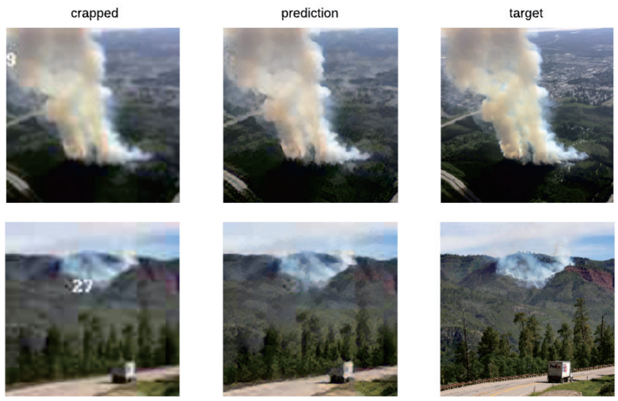

Fig. 7. GAN output. Crapped image, GAN prediction image, and target image.

Step behind the generative adversarial network (GAN), VGG16 based loss network trained U-net help this GAN to catch these features. VGG16 is used here just for simplify the problem, there should be better choices. Because the input to this part is the target and the output of GAN, the $L1$ error of these two images' can be used to compare their features better. The features are grabbed by feature loss network as in paper [14], it is shown in Fig. 1. This L_1 loss is the feature loss or perceptual loss. Then we train the U-net in this part by learner, but the loss function is the feature loss. After we find an appropriate learning rate, we train this model as more as we can in the permission of GPU.

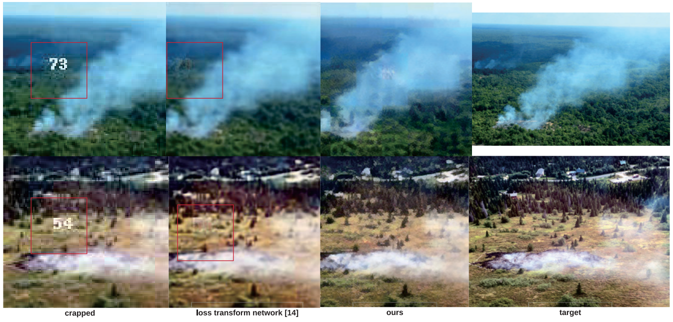

Fig. 8. Comparison result with perceptual loss transfer network.

After we trained this U-net 200 epochs, it shows that the pixels are surely increased and we can recognize the smoke and the forest or grass. By comparing the results with the loss transfer network, the proposed model looks sharper and it nearly removed the number entirely, which means our model get the feature of the image and understand what the missing part should be. Fig. 8 is the comparison result.

For the convenience to the interested readers, the developed code in this paper is available at https://github.com/qiaolinhan/ws-preprocess.

IV. Conclusions and Future Work

In this paper, a loss U-net enhanced GAN is designed to restore the wildfire images for better guidance in forest protection. It grabs the main features by an adaptive cross-entropy form and a loss network. The simulated annealing learning rate is used to help reducing loss. The performance of this model is good and the comparison results showed that it has desired ability of anti-interference because it could even remove the entire number mark in the image. However, during the discriminator design, lack of enough images will not train critic of GAN very well, it is caused by the spectral normalization.

For rigid body object detection, this model performs well, but for smoke or something non rigid, more information are needed. The generated images in this paper can not reflect the dynamical turbulence features. For this reason, our future work will consider the video signal, motion detection or to rebuild the turbulence model of smoke. This could be our smoke foreground segmentation research part to guide fire-line forecasting and wildfire fighting better.

Acknowledgments

Programming of this work is mainly developed on Pytorch and fastai framework, we would like to thank Pytorch developers and fastai discussion for helpful information and layer API design.

The work of this paper is partially supported by the Natural Sciences and Engineering Research Council of Canada, National Natural Science Foundation of China (No. 61833013), and the Key Research and Development Program of Shaanxi Province of China (No. 2020SF-376).

References

[1] C. Yuan, *et al*. "Aerial Images-based Forest Fire Detection for Firefighting Using Optical Remote Sensing Techniques and Unmanned Aerial Vehicles." *Journal of Intelligent and Robotic Systems* vol. 88, no. 2-4, pp. 635-654, 2017.

[2] A. D. Kulkarni." Neural Nets for Image Restoration." *ACM CSC90. Association for Computing Machinery*, pp. 373378, 1990.

[3] M. R. Banham, *et al*. "Digital Image Restoration." *IEEE Signal Processing Magazine* vol. 14, no. 2, pp. 24-41, 1997.

[4] C. H. Shan, *et al*. "Residual Learning of Deep Convolutional Neural Networks for image Denoising." *Journal of Intelligent and Fuzzy Systems* vol. 37, no. 2, pp. 2809-2818, 2019.

[5] G. Coln, *et al*. "Image Restoration using Space-variant Gaussian Scale Mixtures in Overcomplete Pyramids." *IEEE Transactions on Image Processing* vol. 17, no. 1 pp. 27-41, 2007.

[6] L. Bee, *et al*. "Enhanced Deep Residual Networks for Single Image Super-resolution." *IEEE CVPR17*.pp. 136-144, 2017.

[7] G. Huang, *et al*. "Densely Connected Convolutional Networks." *IEEE CVPR17*.pp. 4700-4708, 2017.

[8] S. Li, *et al*. "Imaging Through Glass Diffusers Using Densely Connected Convolutional Networks." *Optica* vol.5, no. 7, pp. 803-813, 2018.

[9] Y. Tai, *et al*. "Memnet: A Persistent Memory Network for Image Restoration." *IEEE ICCV17*. pp. 4539-4547, 2017.

[10] K. Yu, *et al*. "Path-restore: Learning Network Path Selection for Image Restoration." *arXiv preprint arXiv*:1904.10343, 2019.

[11] X. G. Pan, *et al*. "Exploiting Deep Generative Prior for Versatile Image Restoration and Manipulation." *arXiv preprint arXiv*:2003.13659, 2020.

[12] I. J. Goodfellow, *et al*. "Generative Adversarial Nets." *arXiv preprint arXiv*:1411.1784, 2014.

[13] T. Y. Lin, *et al*. "Focal Loss for Dense Object Detection." *Proceedings of the IEEE ICCV17*. pp. 2980-2988, 2017.

[14] J. Justin, *et al*. "Perceptual Losses for Real-time Style Transfer and Super-resolution."

[15] T. Miyato, *et al*. "Spectral Normalization for Generative Adversarial Networks." *arXiv preprint arXiv*:1802.05957, 2018. *ECCV16*, pp. 694-711, 2016.

[16] J. Howard, *et al*. "Fastai: A layered API for Deep Learning." *Information* vol. 11, no. 2, pp. 108, 2020.

[17] A. Buades, *et al*. "A Non-local Algorithm for Image Denoising." *IEEE CVPR05*. Vol. 2, pp. 60-65, 2005.

[18] K. Dabov, *et al*. "Image Denoising by Sparse 3-D Transform-domain Collaborative Filtering." *IEEE Transactions on Image Processing* vol. 16, no. 8, pp. 2080-2095, 2007.

[19] J. K. Choi, *et al*. "An Edge Driven Wavelet Frame Model for Image Restoration." *Applied and Computational Harmonic Analysis* vol. 48, no. 3, pp. 993-1029, 2020.

[20] Y. L. Zhang, *et al*. "Residual Dense Network for Image Restoration." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1-1, 2020.

[21] C. Bertocchi, *et al*. "Deep Unfolding of a Proximal Interior Point Method for Image Restoration." *Inverse Problems* vol. 36, no. 3, 2020.

[22] J. Q. Qiu, and K. Y. Xie. "A GAN-based Motion Blurred Image Restoration Algorithm." *IEEE ICSESS19*, pp. 211-215, 2019.

[23] X. G. Pan, *et al*. "Exploiting Deep Generative Prior for Versatile Image Restoration and Manipulation." arXiv preprint arXiv: 2003.13659, 2020.

[24] L. Loshchilov, *et al*. "Decoupled Weight Decay Regularization." *arXiv preprint arXiv*:1711.05101, 2017.