Adversarial Scene Editing: A Weakly Supervised Study of Wire Detection

Supervisor: Louis Lettry, Student: Zeyu Zhang
CVLAB, School of Computer and Communication Sciences
zeyu.zhang@epfl.ch

Abstract

Over the past few decades, Unmanned Aerial Vehicles (UAVs) have undergone unprecedented development. However, some practical issues still limit its further promotion. One of the most important obstacles is how to guarantee them to fly safely at low altitudes. There are a lot of wires in low-altitude situations that the UAV needs to detect and avoid collisions with high precision. Existing vision-based systems mostly learn a deep neural network to detect wires from the captured image; such models are trained in a supervised manner, which means that pixel-wise annotated training sets are required. However, the acquisition of relevant datasets is usually not easy. Therefore, we propose to use a weakly supervised approach to address this problem: the problem is abstracted into an image-to-image translation task between two domains (wire and wireless). Starting from the Cycle-GAN, we adapt existing models and conduct extensive experiments to explore their feasibility in accomplishing this wire detection task.

1 Introduction

The rapid development of Unmanned Aerial Vehicles (UAVs) has accelerated their application in many fields over the past decades. It is commonly believed that the UAV will be an integral part of future urban civil and military applications. However, many unsolved technical problems restrict its further development. For instance, there is a high damage rate when such vehicles operate at low altitudes with unpredictable obstacles [3], while power lines are regarded one of the most formidable hazards. Thus, great attention has been paid to developing economic and efficient power line detection systems for UAVs. The critical problem of such systems is how to detect the thin wires within receptive fields automatically and accurately. We can categorize the existing power line detection methods into two broad classes: non-vision and vision-based. Regarding the recent advances in deep learning on images, as well as the irreplaceable advantages of camera sensors like low cost and lightweight [16], vision-based approaches are prominently used.

Generally, an intuitive vision-based method for achieving the wire detection objective is to consider it a semantic segmentation task. As one of the first works on the wire detection task using the deep learning approach, [10] employs dilated convolutional layers [19] to build their model and conduct a grid search to find the optimal structure. Because of the lack of pixel-wise labeled training data, they innovatively use a 3D engine to model the power line and synthesize images. Inspired by this study, Zhang et al. [20] further incorporate convolutional features and structured constraints to detect power lines in input images. Although such approaches have obtained promising results, there exists an intractable disadvantage. Due to the characteristics of supervised learning, these methods all require pixel-wise labeled images as the training input [11]. However, a sufficiently large public dataset with pixel-wise annotations is currently unavailable, as collecting and labeling relevant data is exceptionally labor-intensive.

The Report of semester project in CVLAB at EPFL.

In order to overcome the problem of data shortage in a supervised manner, exploring weakly supervised strategies has become a broad consensus. For our wire detection problem, using generative adversarial network-based (GAN-based)methods is a promising direction. The advent of GAN has led to significant progress in various image manipulation tasks. Recent works have demonstrated its feasibility in converting objects from one domain to another. For example, [6] can alter facial attributes like hair orientation, and [21] is capable of changing the artistic style of paintings. Besides, studies like Mask-ShadowGAN [5] can learn to remove objects from input images. These works have an encouraging aspect that the image manipulation is conducted without ground truth supervision, but only with unpaired data from different attribute classes. While the progress in this field is remarkable, several limitations still exist. The proposed models mainly operate on a single centric object within images (like faces) or elements that take up a large percentage of the image (like shadows). Therefore, the gap is still open for manipulating small/thin objects in the image by adopting existing models or designing a new model.

This work moves beyond the limitations stated above and focuses on thin power lines and wires within images. To realize the wire detection objective, we consider it an image manipulation task, precisely, image-to-image translation problem. Consequently, we design the following procedure: Firstly, using a suitable generative model to remove wires from the source input images (this process is conducted in a weakly-supervised manner, paired input data sets are not necessary); Then, subtracting input images with corresponding generated images to obtain wires that we wish to detect. Overall, we can conclude our contribution in three folds –

- A new half wire and half wireless training set is used by leveraging and comparing existing public datasets. This dataset can serve as a new benchmark for future related research;
- We conduct extensive grid searching to find feasible parameters of several commonly used GAN-based models for the image manipulation task and summarize possible directions from the experimental results;
- According to the characteristics of the wire detection problem, we design new loss terms for the objective function. We embed them into existing models to compare and analyze their effects.

2 Preliminaries

2.1 Convolutional neural networks

Convolutional Neural Networks (CNNs) are typical representatives of deep neural models, which contain several convolutional neurons/filters in network structure. CNNs are primarily applied in the field of pattern recognition within images. With the help of convolutional neurons in hidden layers, a CNN can capture the spatially localized features in images. In this work, we mainly use three basic structures for the inference and manipulation purpose over images.

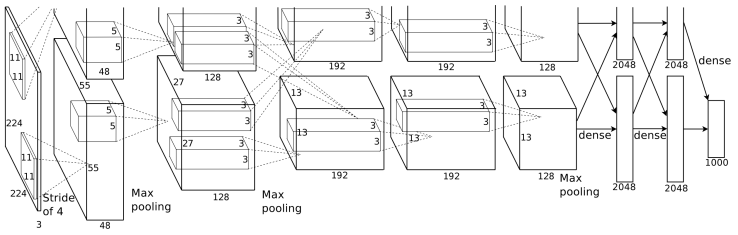

Figure 1: An illustration of the architecture of AlexNet [9].

AlexNet [9] is one of the most popular CNN architectures, which is designed for the image classification task. In 2012, it won the ImageNet Visual Recognition Challenge by a large margin and evoked extensive research on deep learning. As shown in Figure 1, AlexNet consists of different kinds of layers: convolutional layers enforce the model to realize local connectivity patterns within an image; max-pooling layers decrease the model capacity by reducing the dimension of data; fully connected layers aggregate all processed information and give the final output. Moreover, by introducing a few

nonlinear activation functions like ReLU, the model can learn more complicated mapping functions that are suitable for the image inference task.

ResNet [4] While building networks with sufficient convolutional layers yield promising results for many computer vision tasks, researchers are astutely aware that stacking very deep networks will lead to a series of negative consequences. A typical problem is the vanishing/exploding gradients as the number of neural layers increases, which hamper the convergence of training at the beginning. Moreover, the accuracy will get saturated first and then degrade rapidly with the network depth increasing. Accordingly, ResNet is proposed to tackle such issues that happen in very deep networks. Instead of learning unreferenced mappings between the input and output of a couple of convolutional layers, ResNet explicitly reformulates the convolutional layer as learning residual functions with reference to the layers' input (an illustration is shown in Figure 2).

Formally, a building block for a few stacked layers in the ResNet is defined as:

$$\mathbf{y} = \mathcal{F}\left(\mathbf{x}, \{W_i\}\right) + \mathbf{x}, \tag{1}$$

where $\mathbf{x}$ and $\mathbf{y}$ are respectively the input and output vectors of the layers considered, the function $\mathcal{F}\left(\mathbf{x}, \{W_i\}\right)$ represents the residual function that is fitted by the neural layers. With the help of such residual mapping, the model can learn to filter unnecessary information by skipping a few layers. Such shortcut connections between layers can effectively alleviate the issues we stated above. Consequently, building very deep networks to get extra accuracy gain becomes possible.

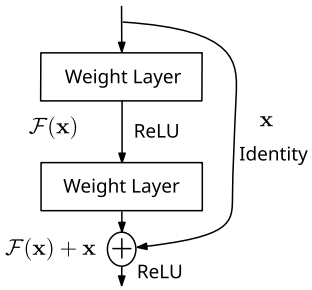

Figure 2: The building block in ResNet.

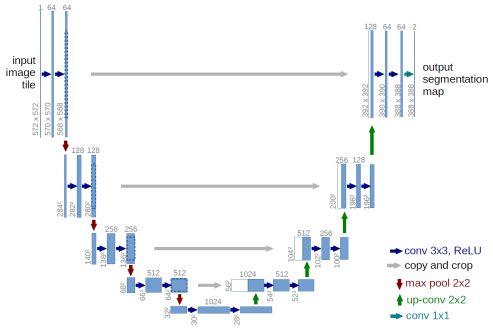

Figure 3: The architecture of U-Net.

U-Net [14] is a fully convolutional network architecture for fast and precise segmentation of images, as there is no fully connected layer in the network. As demonstrated in Figure 3, the U-shaped architecture consists of a contracting path and a symmetric expanding path. The former aims to capture feature correlation within the image, while the latter enables precise localization. The upsampling part also has many feature channels to propagate context information to higher resolution layers. Additionally, the shortcut connections also exist in the U-Net, as it provides the deeper neural layers in expanding path with the original feature information of images, which has a normative effect to avoid the modification deviating too much.

In this study, these three basic neural networks are a foundation for our GAN-based model. ResNet and U-Net are used to generate new images, while AlexNet is mainly for discriminating natural and synthesized images. It should be noted that we did not use the identical models as in the original papers but built similar architectures based on their structural features.

2.2 Generative models

Generative models refer to a series of models used for synthesizing observable data. The key idea of such models is to assign a joint probability distribution between the observations and labeled data sequences. Incorporating with the learned joint probability, the model can generate new data by feeding a random value of the latent variable. Generative models are widely studied in computer vision field to create fake images. In this study, we will mainly exploit the following frameworks:

Generative adversarial network (GAN) [2] Inspired by the game theory, the GAN framework is proposed to train the generative model with an adversarial process, which means another model with a similar capacity supervises the learning of this generative model. Specifically, a generative model G and a discriminative model D are optimized simultaneously during the training, and they are called generator and discriminator, respectively. G endeavors to synthesize fake images based on an input noise variable, as the output should approximate the probability distribution over the target dataset. In contrast, D attempts to distinguish generated fake images and real images from the training dataset. Figure 4 demonstrates the adversarial training process of GAN.

The objective of training process can be formalized as follows:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))], \tag{2}$$

where $p_{\boldsymbol{z}}(\boldsymbol{z})$ is the defined prior on input noise variables, and $p_{\text{data}}(\boldsymbol{x})$ represent the probability that $\boldsymbol{x}$ came from the data. Here we can see the difference between GAN and traditional supervised models: GAN adopts a zero-sum game mode to optimize the generator and discriminator simultaneously rather than optimize the learning process merely based on a loss function. This method of approximating the probability distribution over the training set through a discriminator can effectively help the generator to grasp the global information top-down.

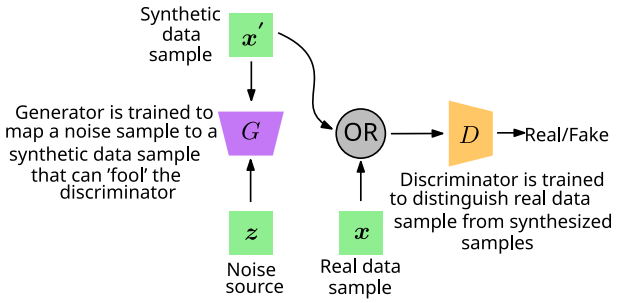

Figure 4: A demonstration of GAN

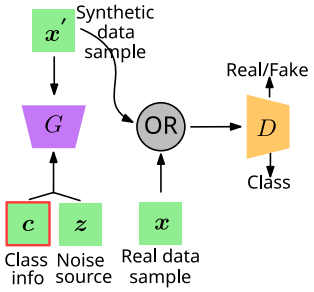

Figure 5: AC-GAN

Auxiliary Classifier GAN [12] As the standard GAN only depends on the discriminator's inference result on real/fake images to improve the generator's performance, the training process is usually unstable, reflecting in the non-convergence of G and D. To alleviate this problem, using side information to augment GAN is considered a promising solution, and one strategy is to supply both the generator and discriminator with class labels to produce class conditional samples. Motivated by this consideration, [12] modify the original GAN with an auxiliary decoder that is tasked with reconstructing class labels. The improved model is called Auxiliary Classifier GAN (AC-GAN). As shown in Figure 5, in an AC-GAN, the generator is fed with a class label $\boldsymbol{c} \sim p_{\boldsymbol{c}}$ and a random noise $\boldsymbol{z} \sim p_{\boldsymbol{z}}$ to generate the synthesized image $X_{\text{fake}} = G(\boldsymbol{c}, \boldsymbol{z})$. Correspondingly, the discriminator attempts to learn both a probability distribution over input images $P(S \mid X)$ and a probability distribution over their class labels $P(C \mid X)$. Thus, the objective function has two main components: the log-likelihood of the correct source L_S, and the log-likelihood of the correct class L_C. Specifically,

$$L_S = E[\log P(S = \text{real} \mid X_{\text{real}})] + E[\log P(S = \text{fake} \mid X_{\text{fake}})] \tag{3}$$

$$L_C = E[\log P(C = c \mid X_{\text{real}})] + E[\log P(C = c \mid X_{\text{fake}})] \tag{4}$$

The discriminator D is trained to maximize $L_S + L_C$, and the generator G is trained to maximize $L_S - L_C$. Thus, the generator will be enforced to generate images belonging to a specific class to fool the discriminator. Although AC-GAN only introduces a slight modification to the standard GAN, it can lead to significant promotion on the quality of generated samples.

Cycle-GAN [21] The image-to-image translation is a problem that aims to transform an image into another relevant image. While determining whether two pictures are relevant is a rather vague concept, this question explicitly refers to converting an image from one domain to an image from

another domain. For example, transform a horse in a picture into a zebra, or transform a painting into another art style. Like [7] [15], using paired datasets to train a generative model is an intuitive and common approach.

Cycle-GAN provides a new way of thinking to solve the image-to-image translation problem. It breaks the limitation of requiring a training set with aligned image pairs and accepts unpaired two image sets as input data. In other words, Cycle-GAN aims to solve the image-to-image translation problem by learning a mapping of the whole of the two image sets rather than matched image pairs. Therefore, it is an unsupervised model that only needs two image sets from different domains. Different from the standard GAN, there are two pairs of generative and discriminative models, and the input to generators is a real image rather than a random variable sample. However, the goal of the two discriminators is still to determine whether the image is synthetic or not.

Because there is no explicit target indication, the generator must first learn the difference between two input datasets. Namely, before making a meaningful conversion, the model needs to be aware of the characteristics and differences between the two different domains. In a large dataset, there may exist numerous feature combinations between different images, which may easily lead to training failure. Cycle-GAN applies cycle consistency to address this problem. The idea is to use transitivity [17] as a way to regularize structured data: after applying the two generators sequentially, the final image should be similar enough to the original one. To fulfill this requirement, a cycle consistency loss is introduced to ensure the generator does not transform images from one domain to another domain that is completely unrelated to the target images.

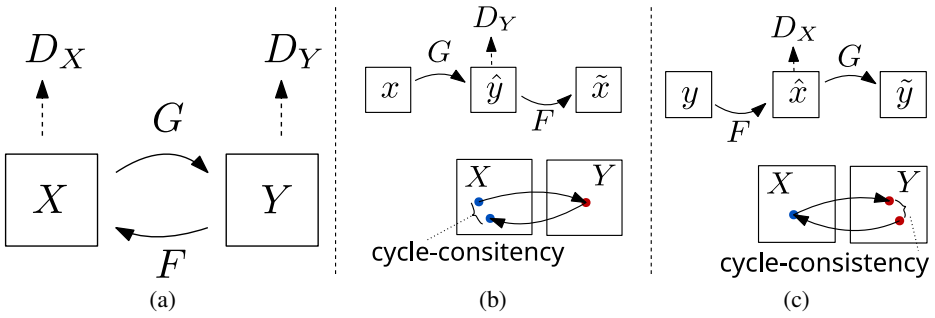

Figure 6: (a) is a brief demonstration of Cycle-GAN. The model contains two mapping functions $G : X \rightarrow Y$ and $F : Y \rightarrow X$ and associated adversarial discriminators D_Y and D_X. D_Y instructs G to translate X into outputs indistinguishable from domain Y, and vice versa for D_X, F, and X. To further regularize the learning, the "cycle consistency losses" is introduced to fulfill the intuition the translation should be invertible by applying G and F sequentially. (b) is the forward cycle-consistency loss; and (c) is the backward cycle-consistency loss.

Formally, the goal of Cycle-GAN is to learn two mutual mappings ($G : X \rightarrow Y$ and $F : Y \rightarrow X$.) between domain X and Y, given training samples $\{x_i\}_{i=1}^{N} \in X$ and $\{y_j\}_{j=1}^{M} \in Y$. Moreover, two discriminators D_X and D_Y are introduced, where D_X is to distinguish between images $\{x\}$ and translated images $\{F(y)\}$; similarly, D_Y is used to discriminate between $\{y\}$ and $\{G(x)\}$. A brief illustration of Cycle-GAN is shown in Figure 6(a).

There are two kinds of loss terms in the objective function – *adversarial losses* and a *cycle consistency loss*. The former is consistent with the standard GAN, matching the probability distribution of generated images to the data distribution in the target domain (as in Equation 2). The latter prevents the learned mappings G and F from contradicting each other. Since there are two mappings to learn, the cycle consistency loss also has two folds: for each image x from domain X, the image translation cycle should be capable of bringing it back to the original image, that is, $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$. This process is called forward cycle consistency (as illustrated in Figure 6(b)). Similarly, a backward cycle consistency (as illustrated in Fogure 6(c)) needs to be satisfied, which can recover an image y from domain Y by applying G and F sequentially, that is, $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$. Then, the cycle consistency loss can be formulated as follows:

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1] \tag{5}$$

Therefore, the full objective is

$$\mathcal{L}(G, F, D_X, D_Y) = \omega_1(\mathcal{L}_{\text{GAN}}(G, D_Y) + \mathcal{L}_{\text{GAN}}(F, D_X)) + \omega_2 \mathcal{L}_{\text{cyc}}(G, F), \tag{6}$$

where ω_1 and ω_2 control the relative importance of the two losses. The training aims to solve:

$$G^*, F^* = \arg\min_{G,F} \max_{D_x, D_Y} \mathcal{L}(G, F, D_X, D_Y) \tag{7}$$

In this work, we first explore the feasibility of using cycle-GAN for our wire detection task. The initiative is natural: constructing two wire/wireless datasets suitable for cycle-GAN is relatively easy compared to the inaccessibility of pixel-wise annotated data. Afterward, we build several GAN-based models by integrating a few features of Cycle-GAN and conduct extensive experiments to analyze their effects.

3 Dataset and problem definition

The first part of this section will briefly introduce and present the dataset we are using. Secondly, we clearly define the problem we are trying to solve.

3.1 Dataset

After searching for existing public datasets, we found the Powerline Image Dataset (Infrared-IR and Visible Light-VL) [1] that is well suited for the wire inspection task in a weakly supervised manner. Images in this dataset originate from videos captured by real aircraft. There are two sub-folders in the dataset, IR and VL, each containing 4000 images (2000 images with wire and 2000 images without wires) with a size of $128\mathbf{x}128$. The IR images are obtained using infrared photography, while the VL images are obtained from a regular camera. In Figure 7, we show some examples from the dataset.

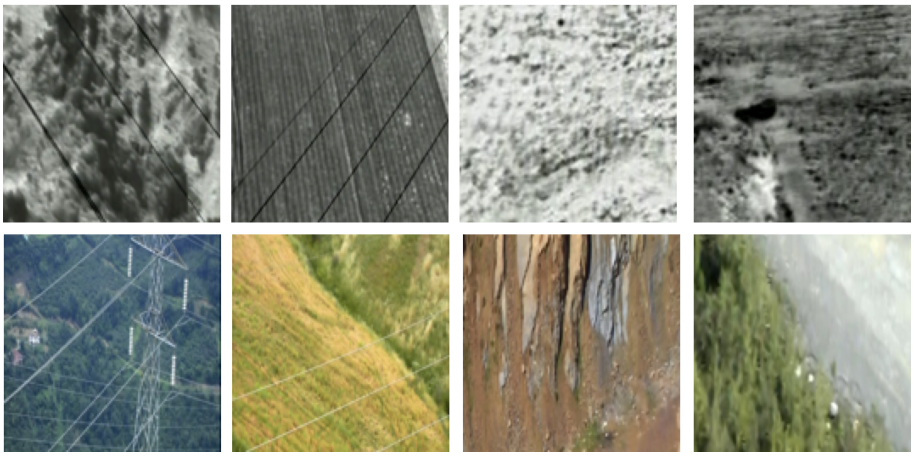

Figure 7: The first row's images are infrared, and the second row's images are visible light; the left half are images with wire, and the right half are wireless images.

Furthermore, this dataset has a great advantage that the collected images cover almost every possible situation: varying from temperatures, weather conditions, light conditions, etc. Evenly, there are several difficult scenes where low contrast causes close-to invisibility for power lines. According to our survey, there are currently no studies using this dataset for weakly supervised learning on wire detection task. We believe that this dataset has the potential to become a benchmark for weakly supervised learning for thin object inspection.

3.2 Problem specification

As previously described, the ultimate goal of this research is to detect wires in pictures. We have a dataset where half of the image set has wires, and the other half does not. To keep the notation consistent in this paper, we refer to them as X and Y, respectively. Because the images in X and Y are not one-to-one matched, we can only do this wire inspection task through a weakly supervised method.

Therefore, we propose to use the following two sequential steps to accomplish this task: First, learn a mapping relation $G : X \rightarrow Y$ that can convert images with wires to those without wires (Importantly, this mapping should only delete the wires from the original image and make no changes to other content); After obtaining the above mapping, for the image $x \in X$, use this mapping to transform it to a new image space $G(x) = \hat{y}$, and $\hat{y} - x$ should be the wires in the image we wish to detect.

According to the definition of the problem, we can find that the key to the problem is how to learn this mapping function G. The trickiest part is that this mapping needs to locate the part of the wire in the picture as precisely as possible, and only do the blending calculation on this part.

4 Learning to remove wires

In this section, we will describe all approaches we attempt to remove wires in the image and their experimental results.

4.1 Cycle-GAN with mask implication

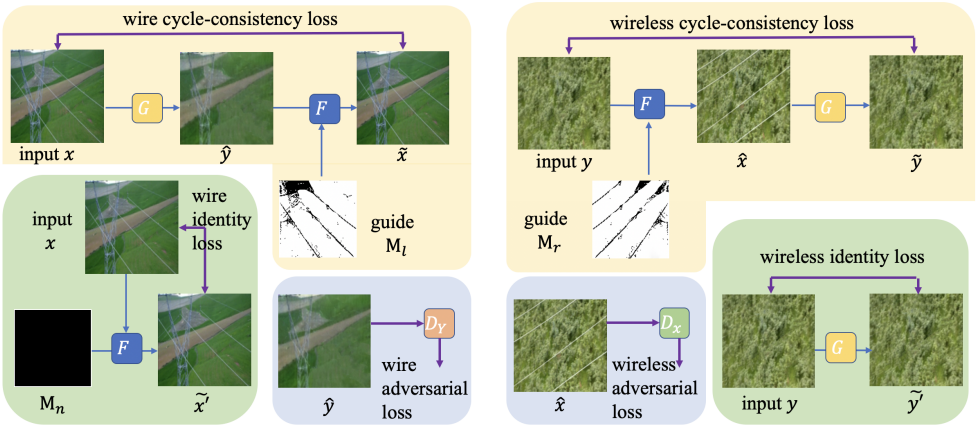

Figure 8: The schematic illustration of the our Cycle-GAN with mask implication, which has two parts: (a) one to learn from real wire images and (b) the other to learn from real wireless images. Each part includes four losses: cycle-consistency loss (yellow), identity loss (green), adversarial loss (blue), and distinct loss (not shown). Besides, G and F denote the generators, which produce the wireless and wire images while D_Y and D_X are the discriminators to determine the whether the generated images are real wireless or wire images. x and y are the real wire and wireless image; $\tilde{x}$ and $\tilde{x}^{'}$ denote the generated wire images while $\tilde{y}$ and $\tilde{y}^{'}$ denote the generated wireless images; M_n, M_l and M_r are the wire masks.

We can think of the process of learning the mapping as an image-to-image translation problem. Again, because the data we deal with are two unpaired datasets, the natural idea is to use cycle-GAN to solve this problem. A regular cycle-GAN model can learn two mappings at the same time. This means that if it succeeds in our goal, the final model can not only remove wires from images with wires but also add wires to images without wires. However, we are faced with the fact that wire pixels occupy only a minuscule fraction of the total number of pixels in an image. Thus, directly employing adversarial learning and cycle-consistency constraints is insufficient to learn the underlying relationship between the wire and wireless domains.

Thus, we need some explicit instructions to assist the model in learning. Inspired by mask-shadowGAN, we also add additional mask information in the learning process to accelerate F learning, which consequently brings positive feedback G for removing wires in the image. Compared to the original cycle-GAN, the input of the generator F includes not only the images from the Y set, but also the mask M obtained from the residual led by G when removing the wires in the image. That is,

$$M = \mathbb{B}\left(\hat{y} - x, t\right), \tag{8}$$

where B indicates the binarization operation, which sets the pixels as one, when their values are greater than the threshold t, otherwise, as zero. The threshold t is obtained by Otsu's algorithm [13], which calculated the optimum threshold to separate wire and non-wire regions by minimizing the intra-class variance. A demonstration of our adapted framework is shown in Figure 8.

In addition, two new terms have been added to the objective function in our implementation. We call the first one *identity loss*, which controls the generator not to change the image from the target set. For example, generator G should leave images from Y without wires intact, so is for F and X:

$$\mathcal{L}_{\text{identity}}\left(G, F\right) = \mathbb{E}_{x \sim p_{\text{data}}\left(x\right)}\left[\|F(x) - x\|_1\right] + \mathbb{E}_{y \sim p_{\text{data}}\left(y\right)}\left[\|G(y) - y\|_1\right] \tag{9}$$

The second new term is called *distinct loss*, as we use it to avoid images being changed too much by the generator G:

$$\mathcal{L}_{\text{distinct}}\left(G\right) = \mathbb{E}_{x \sim p_{\text{data}}\left(x\right)}\left[\|G(x) - x\|_1\right] \tag{10}$$

Thus, the full objective function is:

$$\begin{aligned}\mathcal{L}\left(G, F, D_X, D_Y\right) = \omega_1(\mathcal{L}_{\text{GAN}}\left(G, D_Y\right) + \mathcal{L}_{\text{GAN}}\left(F, D_X\right)) + \omega_2\mathcal{L}_{\text{cyc}}\left(G, F\right) + \\ \omega_3\mathcal{L}_{\text{identity}}\left(G, F\right) + \omega_4\mathcal{L}_{\text{distinct}}\left(G\right).\end{aligned} \tag{11}$$

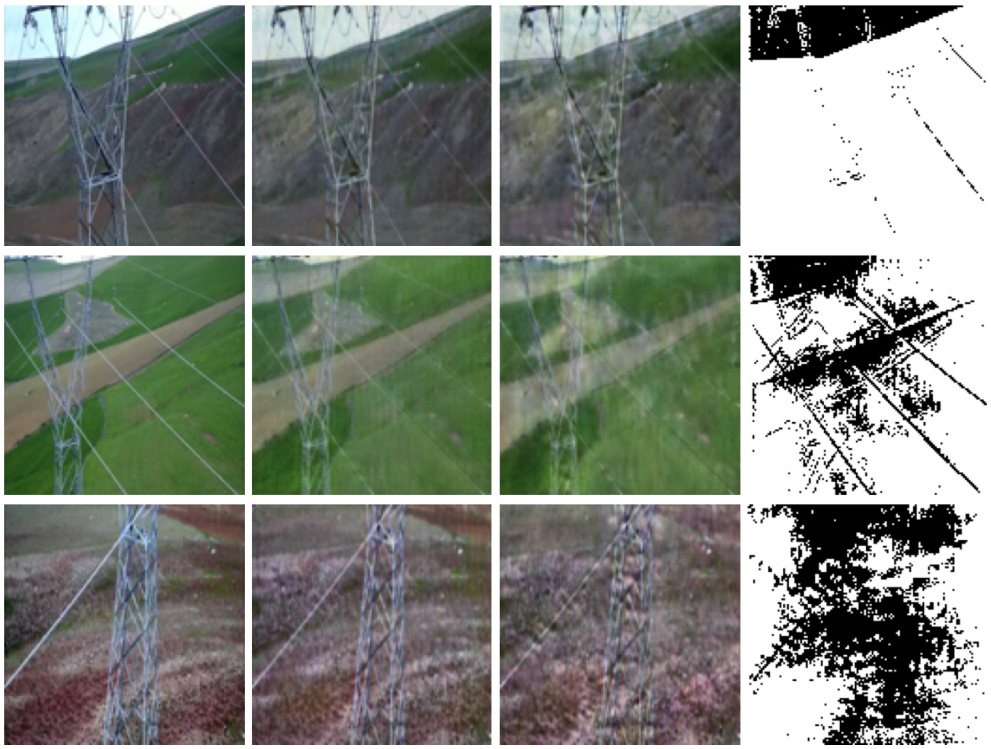

Figure 9: The original image (first column); the generated wireless images at epoch 500 and 1000 (second column and third column), and the wire mask at epoch 500 (last column). Several key hyper-parameters: batch size is 16; $\omega_1 = 1e-3$; $\omega_2 = 1$; $\omega_3 = 1$; $\omega_4 = 0$; generators are optimized ten times than discriminators.

As for the specific network structures used in the model, the two discriminators use a simplified version of AlexNet to distinguish whether the image inputs are real or fake. Besides, we reuse the network

structure designed by [8] for our generators. This network is a combination of ResNet and U-Net, which enables an end-to-end transformation. It includes three convolutional operations, nine residual blocks with the stride-two convolutions, and two deconvolutions for feature map upsampling. In this network, instance normalization [18] is used after each convolution and deconvolution operation. The generator G takes the wire image with the channel number of three as the input, while the generator F receives the concatenation of the wireless image and wire mask as the input.

During training, we conduct extensive experiments to finetune parameters for optimizing the model, including weights for different loss terms, learning rates for discriminators and generators, batch size, number of epochs, etc. In addition, some common tricks are also applied, such as changing the optimizer, adding soft label noise, etc. In this report, we temporarily omit the specific parameter searching details. We also compare the effect of several loss terms: Cycle consistency loss is critical to the completion for a successful training, which can speed up the model to realize wires in the image; Identity loss has a certain auxiliary effect on cycle consistency loss; Distinct loss is useless in this setting, and it does not effectively prevent the image from being excessively blurred, and even affected the smooth progress of training.

We will show the experimental result on a relatively promising set of parameter spaces. As shown in Figure 9: In the early stage of training, the generator can remove wires in some images (when wires are thin and strongly contrasting with the background), but it is not effective for all images, especially for those very thick wires; At the end stage of training, almost all generated images become very blurred. We speculate that this phenomenon occurs because the model recognizes the difference between images in the two domains and tries to convert between domains. Nevertheless, the wires within the training set have several different characteristics, and it is difficult for the model to treat them uniformly. As a result, the phenomenon of mode collapse appears, that is, some latent features that can deceive the discriminator are learned. Overall, based on the early performance of the model, we believe that Cycle-GAN still has the potential to accomplish this task by finding a more optimized parameter combination or expanding the dataset.

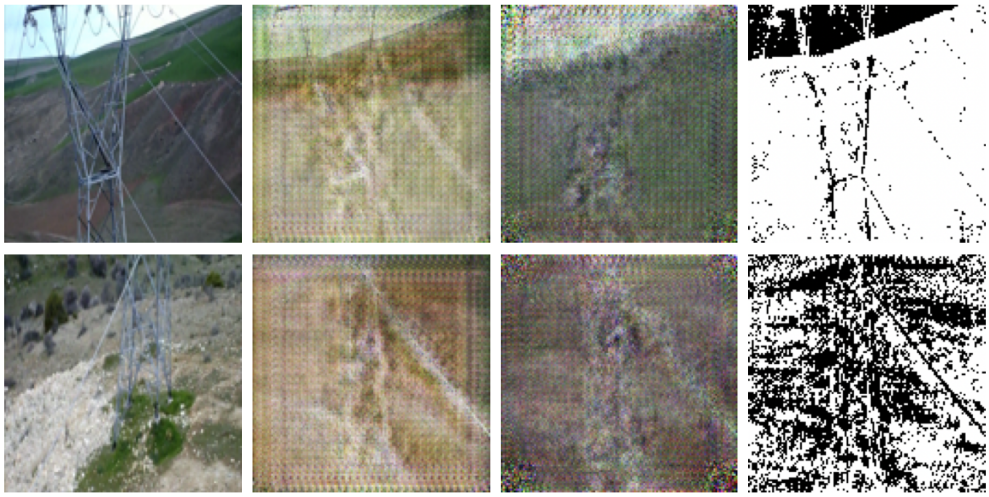

Figure 10: The original image (first column); the generated wireless images at epoch 500 and 1000 (second column and third column), and the wire mask at epoch 1000 (last column).

4.2 Standard GAN

In the previous subsection, we can observe that the cycle-GAN can remove/weak wires from a few images in our dataset. However, while four networks are required to be trained simultaneously in a CycleGAN, it is more challenging to finetune. Because the generator F is not mandatory for our task, we wish to eliminate Cycle-GAN into a simple one for accomplishing our task. Thereby, we adapt the standard GAN as follows: a generator learns the mapping function between X and Y; the discriminator needs to separate the generated images from those from X and Y. This process is relatively simple; thus, we only need to use the adversarial loss and identity loss to guide the training.

However, a frustrating result emerges after many experiments; that is, the model tends to generate meaningless images due to the little guidance information. As shown in Figure 10, the generated images are really fake. Surprisingly, the resulting mask implies that the generator is aware of the location of the wires in the image. This means that using GAN-based models to solve this problem is promising, as long as we have more information to assist the learning process.

4.3 AC-GAN

Based on the above discussion, we find that cycle-GAN has the potential to remove wires in images. However, due to the large number of model parameters, it is difficult to find the optimal parameters. Besides, although standard GAN can be aware of the wires in the picture during the learning process, it would easily fail and produce meaningless outputs. Therefore, we consider finding a compromise way to solve our task.

Ultimately, we choose to leverage the AC-GAN to address this problem. In our model, the purpose of the generator is to learn the mapping function G, whose structure is the same as the one in Cycle-GAN. Differently, the structure of the discriminator is more complicated; it needs to make the authenticity information of the image and classify the image into the correct class meanwhile. There are two parallel structures after the common convolutional layers; one is used for authenticity recognition and the other for classification inference (the structure used to distinguish real/fake images is a bit more complicated). The objective function of the model is adopted from Equation 11, since the basic principle is similar so it is not shown in detail here.

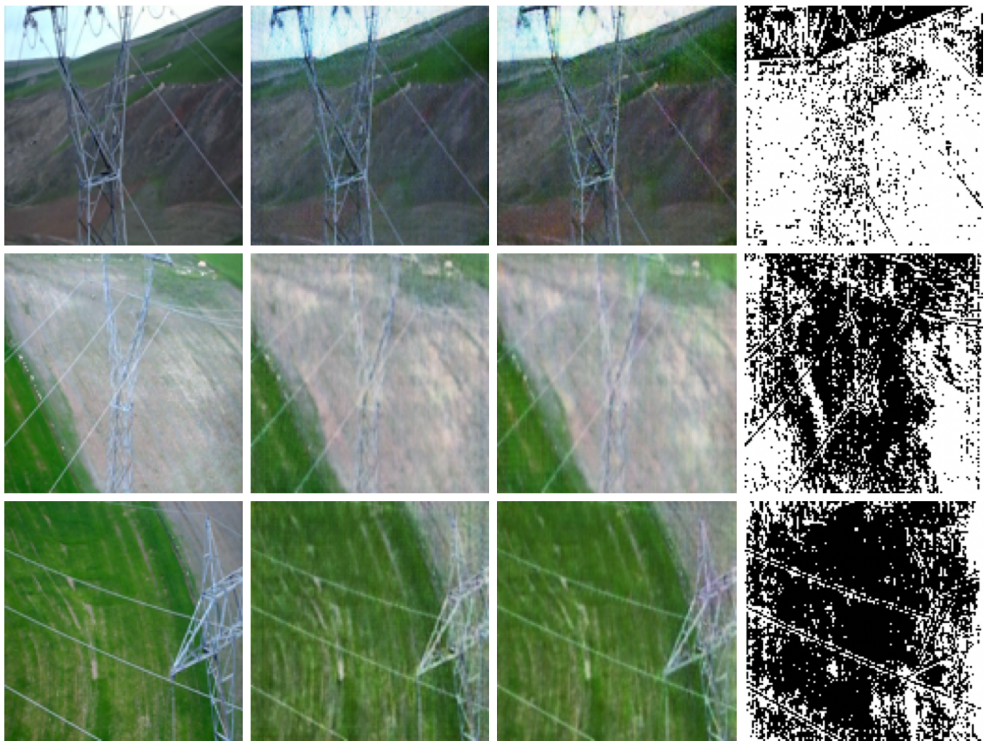

Figure 11: The original image (first column); the generated wireless images at epoch 500 and 1000 (second column and third column), and the wire mask at epoch 1000 (last column). Several key hyper-parameters: batch size is 16; $\omega_1 = 0.5$; $\omega_3 = 10$; $\omega_4 = 5$; generators are optimized four times than discriminators.

Again, we performed several experiments to finetune this model. Here, we present a promising set of results. As can be seen from Figure 11, the generator can weaken the picture to a certain extent. Moreover, it seems to have the ability to remove wires with different characteristics, e.g. thin wires, curve wires, and even very thick wires. Although AC-GAN does not use cycle consistency loss to assist the model in realizing the difference between wire and wireless images, the class information

of images and identity loss largely complement this function. Besides, distinct loss is also useful in this model, while it does prevent the picture from being overly blurred. However, the entire model tends to be conservative, that is, the generator will change the input image as little as possible to deceive the discriminator. Overall, we regard such an AC-GAN based model is also promising to remove wires. To improve the model performance, we can try to increase the discriminator's capacity, or expand the dataset, etc.

5 Conclusions

This study attempts to use a weakly supervised model to conduct the wire detection task. To fulfill this objective, a two-stage procedure is proposed – first learning a mapping function to remove wires in an image; then, the wires are the residual between the original image and the transformed one. Due to the unavailability of paired training data, the mapping function should be learned from two unpaired sets of wire and wireless domains. Thus, we adapt several models for completing this task based on Cycle-GAN. Moreover, we summarize potential directions according to experimental results.

Acknowledgements

Here, I would like to express sincere thanks to my supervisor Louis for his kind help. When I felt lost and confused about the project, he could always give me positive feedback and timely support. I sometimes had questions about life and study, and he also gave me very valuable suggestions. I feel very lucky and honored to be supervised by him.

References

[1] Yetgin Ömer Emre, G Ömer Nezih, et al. Powerline image dataset (infrared-ir and visible light-vl). *Mendeley Data*, 7, 2017.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[3] E Hanna, P Straznicky, and R Goubran. Obstacle detection for low flying unmanned aerial vehicles using stereoscopic imaging. In *2008 IEEE Instrumentation and Measurement Technology Conference*, pages 113–118. IEEE, 2008.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[5] Xiaowei Hu, Yitong Jiang, Chi-Wing Fu, and Pheng-Ann Heng. Mask-shadowgan: Learning to remove shadows from unpaired data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2472–2481, 2019.

[6] Rui Huang, Shu Zhang, Tianyu Li, and Ran He. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In *Proceedings of the IEEE international conference on computer vision*, pages 2439–2448, 2017.

[7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[8] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[10] Ratnesh Madaan, Daniel Maturana, and Sebastian Scherer. Wire detection using synthetic data and dilated convolutional networks for unmanned aerial vehicles. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3487–3494. IEEE, 2017.

[11] Milagros Miceli, Martin Schuessler, and Tianling Yang. Between subjectivity and imposition: Power dynamics in data annotation for computer vision. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW2):1–25, 2020.

[12] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, pages 2642–2651. PMLR, 2017.

[13] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[15] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2017.

[16] Biqin Song and Xuelong Li. Power line detection from optical images. *Neurocomputing*, 129:350–361, 2014.

[17] Narayanan Sundaram, Thomas Brox, and Kurt Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *European conference on computer vision*, pages 438–451. Springer, 2010.

[18] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[19] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[20] Heng Zhang, Wen Yang, Huai Yu, Haijian Zhang, and Gui-Song Xia. Detecting power lines in uav images with convolutional features and structured constraints. *Remote Sensing*, 11(11):1342, 2019.

[21] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.