Generative Adversarial Networks for Visible to Infrared Video Conversion

Mohammad Shahab Uddin
Old Dominion University, muddi003@odu.edu

Jiang Li
Old Dominion University, jli@odu.edu

Chiman Kwan (Ed.)

Original Publication Citation
Generative Adversarial Networks for Visible to Infrared Video Conversion

Mohammad Shahab Uddin and Jiang Li
Department of ECE, Old Dominion University, Norfolk, VA

Abstract

Deep learning models are data driven. For example, the most popular convolutional neural network (CNN) model used for image classification or object detection requires large labelled databases for training to achieve competitive performances. This requirement is not difficult to be satisfied in the visible domain since there are lots of labelled video and image databases available nowadays. However, given the less popularity of infrared (IR) camera, the availability of labelled infrared videos or image databases are limited. Therefore, training deep learning models in infrared domain is still challenging. In this chapter, we applied the pix2pix generative adversarial network (Pix2Pix GAN) and cycle-consistent GAN (Cycle GAN) models to convert visible videos to infrared videos. The Pix2Pix GAN model requires visible-infrared image pairs for training while the Cycle GAN relaxes this constraint and requires only unpaired images from both domains. We applied the two models to an open-source database where visible and infrared videos provided by the signal multimedia and telecommunications laboratory at the Federal University of Rio de Janeiro. We evaluated conversion results by performance metrics including Inception Score (IS), Frechet Inception Distance (FID) and Kernel Inception Distance (KID). Our experiments suggest that cycle-consistent GAN is more effective than pix2pix GAN for generating IR images from optical images.

Keywords: Image Conversion, Generative Adversarial Network, Cycle-consistent Loss, IR Image, Pix2Pix, Cycle GAN

1. Introduction

Image-to-image conversion such as data augmentation [1] and style transfer [2] has been applied to recent computer vision applications. Traditional image conversion models had been investigated for specific applications [3-14]. Since the creation of the GAN model [15], it opened a new door to train generative models for image conversion. For example, computer vision researchers have successfully developed GAN models for day-to-night and sketch-to-photograph image conversions [16]. Two recent popular models that can perform image-to-image translations are Pix2Pix GAN [2] and Cycle GAN [16]. Pix2Pix GAN needs paired images for training whereas Cycle GAN relaxes this constraint and can be trained with unpaired images. In practice, paired images from different domains are often
difficult to obtain. Therefore, Cycle GAN is a better choice for image to image translation where paired images are not available.

IR image datasets are not largely available as compared to optical images. As a result, we face the shortage of data when we train models for object detection in IR domain. This problem can be mitigated by using the Cycle GAN model to covert labelled optical images to IR images. In this chapter, we evaluate two models, Pix2Pix GAN and Cycle GAN, for image conversion from optical domain to IR domain. We used four different datasets to perform the conversion and three metrics including Inception Score (IS), Frechet Inception Distance (FID) and Kernel Inception Distance (KID) to assess quality of the converted IR images.

2. Image to Image Conversion Models

2.1 Generative Adversarial Network

![Figure 1. Structure of Generative Adversarial Network](image)

GAN consists of one generative model and one discriminative model to generate images from noise as shown in Fig. 1. The generator ‘G’ tries to generate images from the input noise ‘z’ as realistic as possible to misguide the discriminator ‘D’ whereas ‘D’ is trained to discriminate the fake image ‘G(z)’ from the real one ‘x’. During training, errors at output ‘D’ are backpropagated to update parameters in ‘G’ and ‘D’, and the following loss function is optimized [15]:

\[
\min \max_{G,D} \mathcal{V}(D; G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]
\]

(1)

where \(x\) and \(z\) represent training data and input noise, respectively. \(p_{data}(x)\) and \(p_z(z)\) are distributions of training data and input noise. The discriminator ‘D’ is trained to minimize the probability of the generated fake image to be real so that it can correctly assign labels to ‘G(z)’ and ‘x’ in Fig. 1. The generator ‘G’ is trained to maximize \(D(G(z))\) or equivalently to minimize \(\log(1 - D(G(z)))\) in equ (1), generating realistic images. Essentially, the generator learns to generate real data’s distribution given by the training dataset. Once the goal is achieved, the generator can be used to generate realistic images by sampling from the learned probability distribution.

2.2 Conditional GAN

GAN can be converted into a conditional model with auxiliary information that is used to impose condition on generator and discriminator [17]. In the conditional GAN model, additional data are fed into the generator and discriminator so that data generation can be controlled. The loss function in conditional GAN becomes [17]:

\[
\min_g \max_D \mathcal{V}(D; G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log D(G(z))|y] + \mathbb{E}_{y \sim p_y(y)}[\log(1 - D(G(z)|y))]
\]
\[
\min_G \max_D V(D; G) = E_{y \sim p_{data}(y)}[\log D(y|x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z|x)))]
\]

(2)

where \(y\) and \(z\) are training data and input noise, respectively. The input noise \(z\) combined with extra information \(x\) generate the output \(G(z|x)\). Fig. 2 shows the diagram of conditional GAN.

**Figure 2.** Architecture of Conditional GAN. Extra information \(x\) is given to both \(G\) and \(D\). The discriminator trains itself to distinguish between real and fake image. The generator trains itself to fool discriminator by generating images similar to real images. Here both \(G\) and \(D\) get \(x\) as input.

### 2.3 Pix2Pix GAN

The Pix2Pix GAN model is built upon the concept of conditional GAN and it has been a common platform for various image conversion tasks. The diagram of Pix2Pix GAN model is given in Fig. 3. Pix2Pix GAN consists of a “U-Net” [18] based generator and a “PatchGAN” discriminator [2]. The “U-Net” generator passes low level information of input image to output image, and the “PatchGAN” discriminator helps capture statistics of local styles. The loss function of pix2pix GAN is:

\[
\min_G \max_D V(D; G) = E_{x,y}[\log D(x,y)] + E_{x,z}[\log(1 - D(x,G(x,z)))] + E_{x,y,z}[||y - G(x,z)||_1]
\]

(3)

Pix2Pix GAN learns to map input image \(x\) and random noise \(z\) to output image \(y\). The generator tries to minimize the loss function while the discriminator tries to maximize the loss function. The \(L_1\) loss between real image and fake one is included to achieve pixel level matching. Pix2Pix GAN had been applied to many applications including edges-to-photo conversion, sketch-to-photo conversion, map-to-aerial photo conversion etc. The main drawback of Pix2Pix GAN is that it needs paired images in both domains for training, which is not always possible in practice.
2.4 Cycle GAN

In many cases, it is difficult to get paired images from different domains. Cycle GAN [16] addressed this challenge by introducing the cycle-consistent loss function as shown in Fig 4. There are two generator G and F in Cycle GAN along with two adversarial discriminator D_x and D_y. X and Y are input domain and target domain, respectively. While D_x helps G to generate images from X domain to Y domain, F is trained to generate images from Y domain to X domain. G: X→Y and F:Y→X are two mappings that are trained in Cycle GAN and these are kept consistent by two cycle-consistency losses. The total loss function of Cycle GAN is given by:

\[
\min_{G,F} \max_{D_x,d_y} \ L(G,F,D_x,D_y) = L_{GAN}(G,D_Y,X,Y) + L_{GAN}(F,D_x,Y,X) + \lambda L_{cyc}(G,F)
\]

where

\[
L_{GAN}(G,D_Y,X,Y) = E_{y\sim\text{pdata}}[\log D_Y(y)] + E_{x\sim\text{pdata}(x)}[\log(1 - D_Y(G(x)))]
\]

\[
L_{GAN}(F,D_x,Y,X) = E_{x\sim\text{pdata}(x)}[\log D_X(x)] + E_{y\sim\text{pdata}(y)}[\log(1 - D_X(G(y)))]
\]

\[
L_{cyc}(G,F) = E_{x\sim\text{pdata}(x)}[||G(F(x)) - x||_1] + E_{y\sim\text{pdata}(y)}[||G(F(y)) - y||_1]
\]

There are two terms in the loss function of Cycle GAN: adversarial losses and cycle-consistency losses. The adversarial losses \(L_{GAN}(G,D_Y,X,Y) + L_{GAN}(F,D_x,Y,X)\) for G: X→Y and F: Y→X mapping, respectively, ensure that target images’ distribution and generated images’ distribution are close. The cycle-consistency loss, \(L_{cyc}(G,F)\), ensures that the two mappings have no contradictions. \(\lambda\) is a weight controlling balance between the two categories of losses.

Cycle GAN has been used in different applications including season transfer, style transfer, etc [16]. In addition, Cycle GAN has resolved the mode collapse problem in training if only the adversarial loss is used [19]. Mode collapse happens when the generator outputs the same image for different inputs. Though other methods [2, 8, 10-2, 20-24] can also offer image-to-image translation with unpaired images, Cycle GAN has become a common platform for many image translation related tasks.

![Figure 4. Overall Architecture of Cycle GAN](image)

3. Experimental Setups

3.1 Datasets

For training Pix2Pix GAN and Cycle GAN, we have used images pairs from the open-source visible and infrared video database from the signal multimedia and telecommunications laboratory at the Federal University of Rio de Janeiro [25]. IR
and visible-light video pairs in the database are synchronized and registered. We utilized 80% of frames in the "Guanabara Bay_take_1" video pair for training and the remaining 20% frames for testing. In addition, we evaluated the trained model on other three image pairs named "Guanabara Bay_take_2", "Camouflage_take_1" and "Camouflage_take_2". Detailed information of the four video pairs are listed in Table 1 and some example pairs are shown in Fig. 5.

**Table 1.** Detailed Information of Video Pairs Used in Our Experiments.

<table>
<thead>
<tr>
<th>Dataset Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Guanabara Bay_take_1    | • Contains scenes of “the Guanabara Bay and the Rio de Janeiro-Niteroi bridge”.
|                         | • Taken during Nighttime.                                                   |
|                         | • Contains 1 scene plane at approximately 500m distance.                   |
| Guanabara Bay_take_2    | • Contains scenes of “the Guanabara Bay and the Rio de Janeiro-Niteroi bridge”.
|                         | • Taken during nighttime.                                                   |
|                         | • Contains 1 scene plane at approximately 500m distance.                   |
| Camouflage_take_1       | • Contains outdoor scenes.                                                  |
|                         | • Taken during bright sunlight.                                             |
|                         | • Contains 2 scene planes at approximately 10m and 300m distances.         |
|                         | • Contains people who are hiding behind vegetation.                         |
| Camouflage_take_2       | • Contains outdoor scenes.                                                  |
|                         | • Taken during bright sunlight.                                             |
|                         | • Contains 2 scene planes at approximately 10m and 300m distances.         |
|                         | • Contains people who are hiding behind vegetation.                         |
3.2 Performance Metrics

3.2.1 Inception Score

Inception score (IS) is widely used for evaluating GANs [26]. IS considers quality and diversity of generated images by evaluating the entropy of probability distribution created by the pre-trained ‘Inception v3’ model on the generated data [27]. A large inception score represents high quality of the generated images. One drawback of the inception score is that it does not consider information in the real images used for training the GAN model. Therefore, it is not clear how the generated images compare to the real training images.

3.2.2 Frechet Inception Distance

Frechet Inception Distance (FID) indicates the similarity between two sets of datasets and is often used for evaluating GANs [28-29]. FID is the Wasserstein-2 distance between feature representations of real and fake images computed by the Inception v3 model [27]. We used the coding layer of the Inception model to obtain feature representation of each image. FID is consistent with the human-judgement of image quality and it can also detect intra-class mode collapse. A lower FID score indicates that the two groups of images are similar so that the generated fake images are of high quality.
3.2.3 Kernel Inception Distance

Kernel Inception Distance (KID) is another metric often used to assess quality of GAN generated images relative to real images [30]. KID first uses the Inception v3 model to obtain representations of generated images. It then calculates the squared maximum mean discrepancy (MMD) between the representations of real training images and generated images. KID score is also consistent with human judgement of image quality. A small KID value indicates high quality of the generated images.

4 Results

4.1 Testing Results on “Guanabara Bay_take_1”

We trained the Pix2Pix GAN and Cycle GAN on 80% of the frames in “Guanabara Bay_take_1” video pair and tested the trained models on the remaining 20% frames. Some visible and IR images that we have used for training are shown in Fig. 5. After training, we applied both models to the testing frames and Fig. 6 shows some generated IR images. By visual inspection, Cycle GAN can generate better results than Pix2Pix GAN does. In addition, we observe that IR images generated by Cycle GAN are similar to the real IR images. Table 2 lists the quantitative performance metrics of the generated images by the two models. Cycle GAN outperforms Pix2Pix GAN in terms of all the metrics including IS, FID and KID on this dataset.

![Generated IR Images by Pix2Pix GAN](a)
Figure 6. Fake IR Images generated by Pix2Pix GAN and Cycle GAN from the visible images in the Guanabara Bay\_take\_1 dataset.
4.3 Testing Results on “Camouflage_take_1” and “Camouflage_take_2”

We have applied the trained models to “Camouflage_take_1” and “Camouflage_take_2” datasets and results are shown in Figs. 8 and 9. Both models did not generate good quality IR images though the quantitative metrics as shown in Table 2. Cycle GAN is slightly better than Pix2Pix GAN. One possible reason is that the data in the two sets have different distributions as those in the training data, making both models failed.
Figure 8. Fake IR Images generated by Pix2Pix GAN and Cycle GAN from the visible images of Camouflage\_take\_1 dataset.
b) Generated IR Images by Cycle GAN

Figure 9. Fake IR Images generated by Pix2Pix GAN and Cycle GAN from the visible images of Camouflage take_2 dataset

Table 2. Evaluation Metrics on Generated IR Images of Different Datasets using Pix2Pix GAN and Cycle GAN.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Guanabara Bay_take_1</th>
<th>Guanabara Bay_take_2</th>
<th>Camouflage take_1</th>
<th>Camouflage take_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>IS Score</td>
<td>PixPix GAN</td>
<td>Cycle GAN</td>
<td>PixPix GAN</td>
<td>Cycle GAN</td>
</tr>
<tr>
<td>FID</td>
<td>2.70</td>
<td>2.88</td>
<td>1.85</td>
<td>3.61</td>
</tr>
<tr>
<td>KID</td>
<td>0.90</td>
<td>0.84</td>
<td>2.33</td>
<td>1.12</td>
</tr>
</tbody>
</table>

4. Conclusion

In this chapter, we have investigated visible-to-IR image conversion using Pix2Pix GAN and Cycle GAN. Cycle GAN is a better model than Pix2Pix GAN and both can generate good visual quality IR images based on visible images, if training data and test data are similar. Overall, IR images generated by Cycle GAN have sharper appearances and better quantitative performance metrics than those by Pix2Pix GAN. However, if testing data have significant distribution shift as compared to training data, both models cannot generate quality IR images. Therefore, our recommendations are 1) Cycle GAN appears to be a better tool to convert optical images to IR images if training and testing datasets have similar distributions and 2) Both models are sensitive to distribution shift and additional techniques are needed to address the challenge.

References

[4]


