Huazhong University of Science and Technology

Infrared image period extension algorithm based on StarGAN

Authors:
Yanling Hua

Instructors:
Zhengrong Zuo

January 11, 2025

Contents

1 Introduction

Infrared radiation exists everywhere in the world. All objects in nature with a temperature above absolute zero emit infrared radiation, and the higher the surface temperature, the stronger the infrared radiation. Infrared radiation is a type of electromagnetic wave with a wavelength range between 0.75 μm and 1000 μm. Based on its wavelength range, the infrared spectrum can be divided into three bands: near-infrared 0.7 μm to 2.5 μm, mid-infrared 2.5 μm to 25 μm, and far-infrared 25 μm to 100 μm. The imaging performance of infrared images is primarily influenced by the scene temperature, the wavelength range of the infrared imaging equipment, and the atmospheric transmission medium. This section focuses on analyzing the effects of temperature and wavelength on infrared imaging.

Planck's law is one of the fundamental laws describing infrared thermal radiation. It quantitatively analyzes the relationship between the spectral radiance of a blackbody and temperature and wavelength. The formula is as follows:

$$M_{b\lambda} = \frac{c_1}{\lambda^5} \cdot \frac{1}{e^{\frac{c_2}{kT}} - 1} \tag{3-1}$$

where $M_{b\lambda}$ represents the spectral radiance, c_1 is the first radiation constant, c_2 is the second radiation constant, λ is the wavelength, T is the temperature, and k is the Boltzmann constant. The constants are $c_1 = 3.7413 \times 10^{-16}\,\mathrm{Wm}^2$, $c_2 = 1.43879 \times 10^{-2}\,\mathrm{mK}$, and $k = 1.380662 \times 10^{-23}\,\mathrm{J/K}$.

Figure 1 shows the relationship between the spectral radiance of a blackbody and temperature and wavelength. From the figure, we observe that the spectral radiance of a blackbody under different temperatures increases sharply with wavelength, reaches a peak, and then gradually decreases. As the temperature increases, the wavelength corresponding to the peak decreases. According to Wien's displacement law, the wavelength of the peak spectral radiance is inversely proportional to the blackbody temperature. By integrating over all wavelengths from 0 to infinity, we obtain the relationship between the total radiance of a blackbody and temperature, known as the Stefan-Boltzmann law:

$$M_0(T) = \sigma T^4 \tag{3-2}$$

where $\sigma = 5.6694 \times 10^{-8}\,\mathrm{W/(m^2K^4)}$. The Stefan-Boltzmann law indicates that the total radiance of a blackbody is proportional to the fourth power of its temperature.

The above analysis is based on the blackbody, which is an ideal theoretical model. To apply the physical analysis to real-world scenarios, it is necessary to quantify the relationship between blackbody radiation and actual objects. Emissivity is a physical quantity that quantifies the ratio of the radiative power of a real object to that of a blackbody. A lower

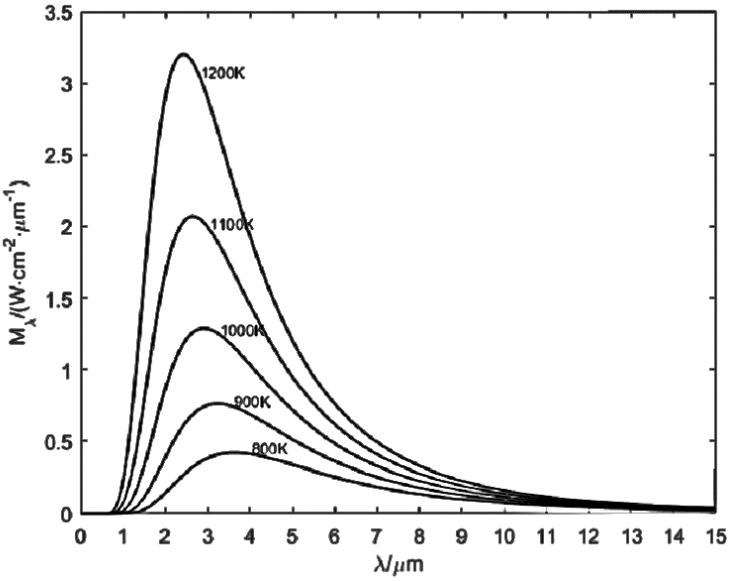


Fig. 1: Relationship between blackbody spectral radiance, temperature, and wavelength

emissivity value indicates a greater difference between the radiation of the object and the blackbody. Thus, knowing the emissivity of any target object allows us to use blackbody radiation theory to analyze its radiation. Different objects have different emissivities due to their physical properties. Generally, the emissivity of metals is directly proportional to temperature, while the emissivity of non-metals is inversely proportional to temperature. Additionally, the emissivity of objects decreases with increasing infrared wavelength.

The heterogeneous infrared image generation algorithm based on multi-receptive field feature fusion Pix2pix can achieve the conversion from visible light images to infrared images of a specific time period. To efficiently generate infrared images for a large number of different time periods, this paper investigates the time-period expansion of infrared images based on Generative Adversarial Networks (GANs). Since it is difficult to obtain paired infrared images for different time periods in the time-period expansion task, this research is based on a network capable of performing unpaired image-to-image translation—StarGAN.

2 Related Work

2.1 Infrared Image Generation Methods Based on Simulation Models

Infrared image generation methods based on simulation models are widely adopted for infrared image simulation. Research on infrared simulation technology abroad began in the 1930s and 1940s. In the early stages, the focus was on modeling the surface temperature distribution of objects, thereby establishing thermal models for these objects.

In 1980, Jacobs developed a surface temperature modeling approach based on a simple one-dimensional thermal model [1]. In 1985, Ben-Yosef et al. conducted radiation statistics for the 8-12 μm infrared band to establish an empirical model [2]. By 1987, Biesel et al. implemented a simulation method capable of real-time generation of highly accurate infrared images [3], creating an infrared scene database. In 1994, Gambotto et al. proposed an infrared simulation algorithm combining segmentation algorithms and regional analysis to simulate ground surface temperatures and calculate corresponding infrared thermal images, enabling all-time infrared temperature model calculations [4].

Research into infrared image simulation expanded significantly in the 1990s. In 1992, Cathcart integrated scene geometric models, infrared radiation models, and computer graphics techniques to render high-resolution synthetic images [5]. In 1996, Hyun-Ki et al. introduced an infrared radiation model that combined target internal thermal models with background environmental temperature distributions [6]. By the late 20th century, Poglio et al. developed a simulation framework based on 3D target models, capable of generating high spatial resolution infrared images for the 3–14 μm band [7].

With the increasing computational power of computers, the 21st century has seen the emergence of many mature infrared simulation platforms, greatly simplifying the simulation process and shortening simulation cycles. In 2003, Lockheed Martin UK developed the CAMEO-SIM simulation software [8], capable of synthesizing infrared images in various environments using geometric models, infrared radiation models, and atmospheric models. Currently, the Vega software package by MultiGen Paradigm is one of the most widely used infrared simulation tools. This software integrates numerous precise material data samples, radiation models, and atmospheric models, enabling infrared simulation across diverse environments and time periods.

In 2007, the French company OKTAL-SE developed SE-WORKBENCH [9, 10], a simulation platform capable of modeling environments for multiple sensor devices, including electro-optical and lidar systems. Its specialized infrared module achieves high-fidelity infrared image simulation. In the following years, researchers began leveraging the powerful

parallel capabilities of GPUs in infrared simulation systems. In 2011, Mielikainen et al. developed a high-speed, high-performance infrared atmospheric sounding interferometer (IASI) based on GPUs [11], which could operate on GPU-equipped computers.

Compared to infrared image simulation technology abroad, China's research in this field started relatively late, and the overall level lags behind that of other countries. However, with increasing national investment in the domain of infrared simulation, significant progress has been made in recent years.

Initially, Chinese researchers focused on simulating the infrared radiation distribution of natural scenes. In 1997, Zhang Jianqi et al. established a model of surface infrared radiation characteristics based on surface radiation properties and the thermal balance equation [12]. In 2000, Wang Zhangye et al. calculated the infrared radiation intensities of buildings' rooftops, walls, glass, asphalt roads, and other objects during different periods based on meteorology and heat transfer theory [13]. In 2002, Xuan Yimin et al. developed an infrared thermal imaging simulation method based on visible images, using a Markov random field segmentation model to calculate thermal radiation characteristics of materials and generate simulated infrared thermal images [14]. In 2010, Chen Shan et al. proposed a method for simulating infrared textures of background scenes. They used visible light images and generated infrared images for different times based on scene temperature models, infrared radiation models, and gray mapping models [15]. In 2015, Zhou Qiang et al. established a grayscale mapping relationship from visible light images to near-infrared images based on radiation calibration and scene emissivity [16]. Although this method achieved fast simulation speeds, the realism of the infrared texture was low. In 2017, Yang Yibin et al. proposed an infrared image simulation method based on infrared grayscale interpolation and modulation. This approach calculated grayscale distributions for other time periods using heat transfer theory [17], which is suitable for simulating simple infrared scenes.

Since the 21st century, China has gradually shifted to researching complete infrared simulation systems. In 2000, Wu Yaping and Zhang Tianxu analyzed the interaction among 3D scene models, atmospheric transmission models, and material physical thermal properties during infrared image simulation. Using Vega software, they completed multi-band infrared image simulations [18]. In 2007, Da Bangyou and Sang Nong used Vega software to simulate images of different materials under various time periods, weather conditions, and seasons. By solving the relationship between simulation results and infrared radiation values, they obtained radiation data as the mean values for corresponding materials [19]. In 2016, Zhong Ming carried out secondary development based on SE-Workbench software, automating the material mapping process and reducing excessive manual intervention [20].

The aforementioned infrared image generation methods based on simulation models rely on physical models of infrared imaging. They are highly interpretable and provide some

degree of physical credibility. However, they involve complex intermediate physical parameters, require significant manual intervention, and feature limited coupling between target and background infrared radiation calculations. As a result, these methods cannot achieve rapid and large-scale infrared image generation.

2.2 Infrared Image Generation Methods Based on Deep Learning

The generation of heterogeneous infrared images and infrared image time-extension can be considered as a mapping process between different image domains. This paper models the image generation task as an image translation problem. Deep learning has been a research hotspot in recent years, and foreign researchers have applied deep networks to perform some image translation tasks in the visible light image domain. In 2016, Gatys et al. utilized convolutional neural networks (CNNs) to achieve artistic style transfer of visible light images [21], but their method suffered from low efficiency and conversion distortion. In the same year, Li Feifei et al. proposed a real-time image style transfer model based on a perceptual loss function, leveraging intermediate feature layers of the VGG-16 network to calculate perceptual losses [22].

Generative adversarial networks (GANs) have been a popular topic in deep learning, achieving impressive results in image-to-image translation tasks. Until recently, generated images were of low quality and resolution. However, this changed with the introduction of Pix2pix, a revolutionary development in the field of image translation. In 2016, Isola et al. proposed the Pix2pix framework [23], which used adversarial loss between a generator and a discriminator combined with L1 loss to handle paired visible light image translation tasks. Subsequently, to address high-resolution image translation tasks, Wang et al. proposed the Pix2pixHD network [24], which employed multi-level generators and multi-scale discriminators, achieving better results.

For unpaired image-to-image translation tasks, Zhu et al. proposed the CycleGAN network in 2017 [25], which adopted dual generator networks and a cycle-consistency loss to enable conversion between unpaired image domains. In the same year, Kim et al. introduced DiscoGAN [26], and Yi et al. proposed DualGAN [27], both employing network architectures similar to CycleGAN and achieving promising results in tasks like image colorization and facial content editing. In 2018, Liu et al. proposed the UNIT network [28], combining a variational autoencoder with a generator. Each image domain had its encoder and generator networks, and the encoders' outputs were assumed to follow a shared distribution, enabling cross-domain image distribution learning. In 2019, Huang et al. introduced MUNIT [29], which employed feature disentanglement to encode images into content codes independent of image domains and style codes specific to image domains. This framework generated images

in different domains by combining content codes with style codes from different domains. However, these frameworks could only learn mappings between two image domains at a time, limiting scalability when dealing with multiple domains. To address this, Choi et al. proposed the StarGAN network in 2018 [30], encoding domain information into the generator input, enabling transformations between multiple image domains using a single generator. This algorithm achieved notable success in facial attribute editing tasks.

The aforementioned deep network architectures were primarily applied to visible light images, with datasets often consisting of well-aligned facial images and relatively simple distributions. However, the infrared datasets used in this study exhibit significant variations in scene content, differing greatly from the distribution of visible light image datasets, leading to notable performance gaps in practical applications.

In recent years, some infrared image generation methods based on GANs have emerged. In 2019, Xie Jiangrong et al. [31] proposed using the DCGAN network for the random generation of simple infrared target images. In the same year, Feng Yunfei investigated the task of infrared band extension and applied the CycleGAN network to convert mid-wave infrared images to long-wave infrared images [32].

3 Method

3.1 Time Period Expansion of Infrared Images Based on StarGAN

3.1.1 StarGAN Network

Infrared imaging is mainly related to temperature, wavelength, and the transmission medium. The transformation relationship between infrared images at different time periods is not only dependent on temperature but also on wavelength and atmospheric transmission factors. Equations (1) and (2) represent the imaging factors of two infrared images at different time periods:

$$I_{T_1} = f(T_1, \lambda_1, \epsilon_1, \theta_1) \tag{1}$$

$$I_{T_2} = f(T_2, \lambda_2, \epsilon_2, \theta_2) \tag{2}$$

Where I_{T_1} represents the infrared image at time period 1, I_{T_2} represents the infrared image at time period 2, λ_1, T_1, and ϵ_1 are the influencing factors for time period 1, while λ_2, T_2, and ϵ_2 are the influencing factors for time period 2. λ represents wavelength, T represents temperature, ϵ represents atmospheric transmission effects, and θ represents other influencing factors.

In this paper, infrared images from three different time periods are selected to represent their respective infrared radiation characteristics: infrared images at 5 AM, 2 PM, and 7 PM. The infrared image dataset is captured using the same imaging equipment, so the wavelength and atmospheric environment remain relatively constant. Based on the above analysis, if we can obtain a mapping function that satisfies Equation (3), we can use it to expand infrared images across different time periods. Therefore, the feasibility of time period expansion depends on whether we can fit the mapping function between time period 1 and time period 2 images.

$$I_{T_1} = F(I_{T_2}) \tag{3}$$

This paper models the time-period expansion of infrared images as an image translation problem between different time-period infrared image domains. To solve the problem of requiring paired datasets, many scholars have proposed image translation models for unpaired datasets: CycleGAN [25], DiscoGAN [26], and DualGAN [27] adopt a dual generator structure, while using a cycle-consistency loss function to preserve key features of the source image domain and reduce unreasonable mapping relationships. UNIT [28] and MUNIT [29], based on image information decoding ideas, combine VAE and GAN networks to decode image information into content features and domain-specific style features, enabling one-to-one and many-to-many image generation. However, in the aforementioned models, a single generator can only learn the mapping relationship between two image domains at a time. When performing image translation across multiple image domains, these models face limited robustness and scalability. As shown in Figure 2(a), to learn all transformation mappings between k image domains, the models must train $k(k-1)$ generators, and each time, only two image domains are used for training, which prevents the learning of global features for the entire dataset.

To improve the algorithm's efficiency, the proposed method enables mutual translation between multiple image domains using a single model. As shown in Figure 2(b), this framework adds target domain image period information at the input side, and a single generator network learns the mapping relationship between infrared images at different time periods. This unified model architecture allows for the transformation relationships between multiple image domains to be obtained while training a single network, enabling the flexible conversion of input images into any required target domain image.

StarGAN, proposed by Choi et al. [30], is a network designed for facial attribute editing. The network allows for multi-attribute facial image editing using a single generator by adding target facial attribute information to the input. Unlike other models that learn fixed transformation relationships between two image domains, StarGAN combines image and domain information as input to the generator, allowing flexible conversion of the image to the

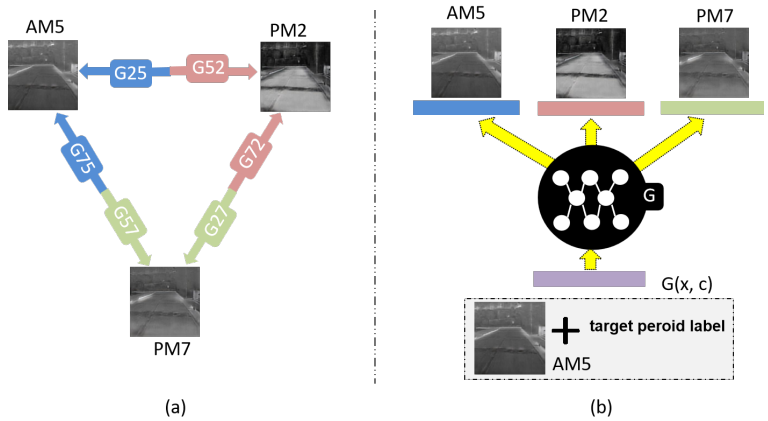


Fig. 2: Comparison of Model Architectures

corresponding domain. For domain information encoding, StarGAN uses one-hot encoding to represent domain information. Based on the previous analysis, StarGAN is well-suited for the task of infrared time period extension. Therefore, in this study, StarGAN is used as the basic framework for the algorithm. The model architecture based on the StarGAN network is shown in Figure 3, where the network consists of a generator network G and a discriminator network D. The input to the generator network is the concatenation of the time period encoding information (green box) and the input infrared image. Meanwhile, the discriminator network adds a time period classification branch.

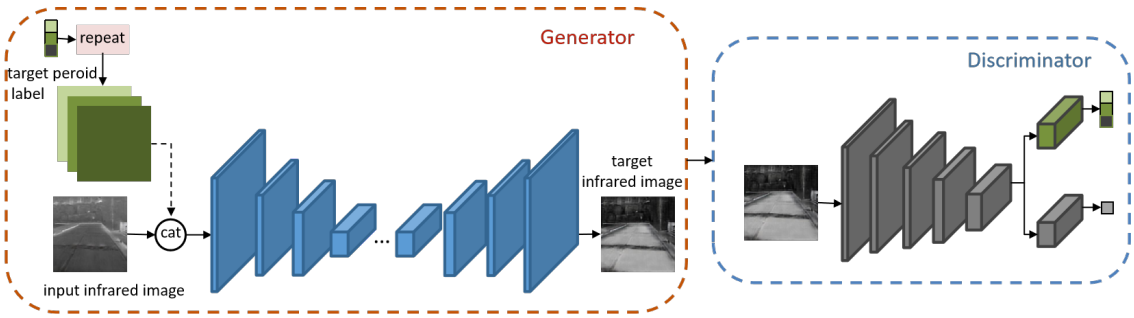


Fig. 3: StarGAN Network Model Structure

During the training process, StarGAN randomly generates target domain time period labels and trains the model to flexibly convert the input infrared image into the target time period domain. Thus, in the testing phase, the image can be converted to any desired target time period infrared image by controlling the target time period label. The training process

of the StarGAN-based infrared time period extension algorithm is illustrated in Figure 4, where the top of the image shows the time period information and its corresponding encoding method:

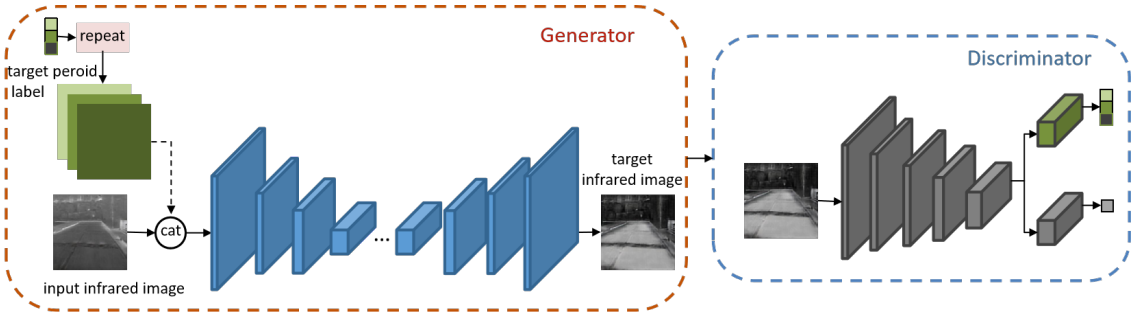


Fig. 4: StarGAN Training Process for Infrared Time Period Extension

- **Training the Discriminator Network (D):** Real infrared images and those generated by the generator are input into the discriminator network. The discriminator learns to distinguish real infrared images from fake ones and classifies real infrared images into the corresponding time period domain (100).

- **Training the Generator Network (G):** Mapping from the source domain to the target domain. The target domain time period encoding label $label_r$ (010) is randomly generated. This label is spatially replicated and concatenated with the input image, and fed into the generator network to generate the corresponding time period infrared image.

- **Training the Generator Network (G):** Mapping from the target domain to the source domain. The generator network attempts to reconstruct the original image by taking the original infrared image with its time period encoding label. The generated image and the original image's time period encoding information $label_s$ are concatenated and input into the generator to produce the reconstructed original infrared image x^{rec}.

- **Training the Generator Network (G):** "Deceiving" the discriminator by generating target domain infrared images that are indistinguishable from real infrared images. The infrared image x' generated by the generator is input into the discriminator network. The discriminator classifies the image as real or fake and outputs the corresponding time period encoding information. At this point, the generator aims for the

discriminator to classify the image as "real" and correctly output the target domain time period encoding $label_r$ (010).

3.1.2 StarGAN Model Architecture

Generator Structure. The generator network structure used in this study is shown in Figure 5. The yellow dashed line represents the encoder, the purple dashed line represents the decoder, and the black dashed line represents the transmission layer. The encoding part consists of three down-sampling convolution blocks that reduce the resolution of feature maps. The transmission layer consists of six residual blocks, while the decoding part recovers the original resolution through three up-sampling convolution blocks. To generate infrared images for multiple time periods, the time period encoding is added to the network's input. The original time period encoding uses a one-hot encoding scheme. The original time period encoding is extended in the 3rd and 4th dimensions and concatenated with the input image along the 2nd dimension before being fed into the generator. By transforming the original time period encoding, the network's output can be controlled. Figure 5 shows the generator network structure, which demonstrates how an infrared image at 5 AM is transformed into infrared images for 2 PM and 7 PM using two different time period encoding information.

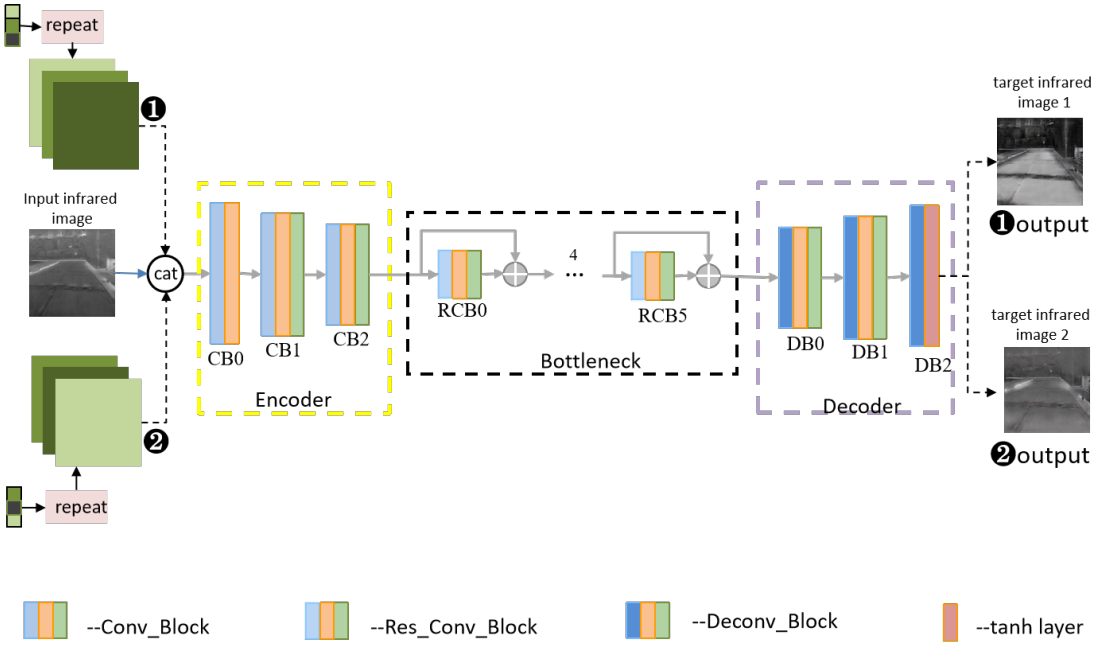


Fig. 5: StarGAN Generator Network Architecture

The structures of down-sampling convolution block (Conv_Block), residual convolution block (Res_Conc_Block), and up-sampling convolution (Deconv_Block) are illustrated in Figure 6, Figure 7, and Figure 8, respectively. Regarding the normalization layer choice, the generator network uses the Instance Normalization (IN) layer, which has shown good experimental results in image style transfer tasks. Batch Normalization (BN) is widely used in convolutional neural networks (CNNs), as it normalizes the data for each batch during training, ensuring consistent data distribution and better adaptation to the overall data distribution. However, in image translation tasks with unpaired datasets, the generated image results primarily rely on the corresponding image instance, and the data distribution in the same batch is less correlated. Therefore, the IN layer is used to normalize each image instance separately. Additionally, residual convolutional neural network (CNN) structures are used in the transmission layer, where the input and output of convolution blocks are added together. This structure helps accelerate the network's learning speed and mitigates the vanishing gradient problem during backpropagation.

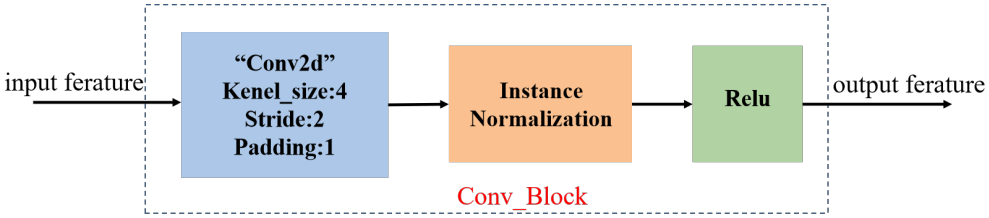


Fig. 6: StarGAN Encoder Submodule Network Architecture

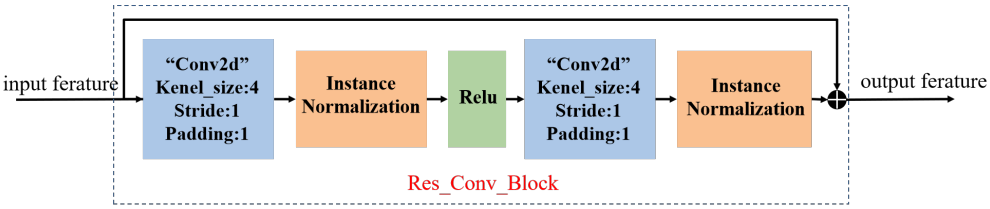


Fig. 7: StarGAN Transmission Layer Submodule Network Architecture

The parameters of the G-network are shown in Table 1. The convolution layer parameters (f, k, s, p) represent the number of filters, filter size, stride, and padding size, respectively. The input and output feature parameters (b, c, h, w) represent batch size, number of channels, feature map height, and feature map width, respectively. From the table, it can be seen that the encoder performs down-sampling by setting the stride of the convolution layers to 2. The transmission layer consists of two convolution layers per submodule, ensuring that the

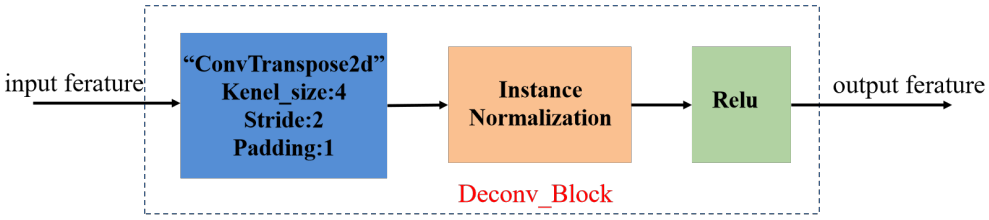


Fig. 8: StarGAN Decoder Submodule Network Architecture

feature resolution remains the same. The image resolution is progressively restored through the decoder's deconvolution layers, ensuring that the output resolution matches the input resolution.

Discriminator Structure. For a given input image and a target time period label, the goal of time period extension is to transform the input image into an output image with the features of the target time period domain. To ensure that the generated image meets the domain conditions of the corresponding time period, the discriminator network (D) adds an auxiliary classifier that performs time period classification on the input image. During the training of both the generator and discriminator networks, a time period classification loss function is incorporated. The specific implementation details can be found in the next section. Additionally, the D network retains an output path for distinguishing between real and fake images. The architecture of the D network used in this study is shown in Figure 9, and the ConvBlock submodule is illustrated in Figure 10.

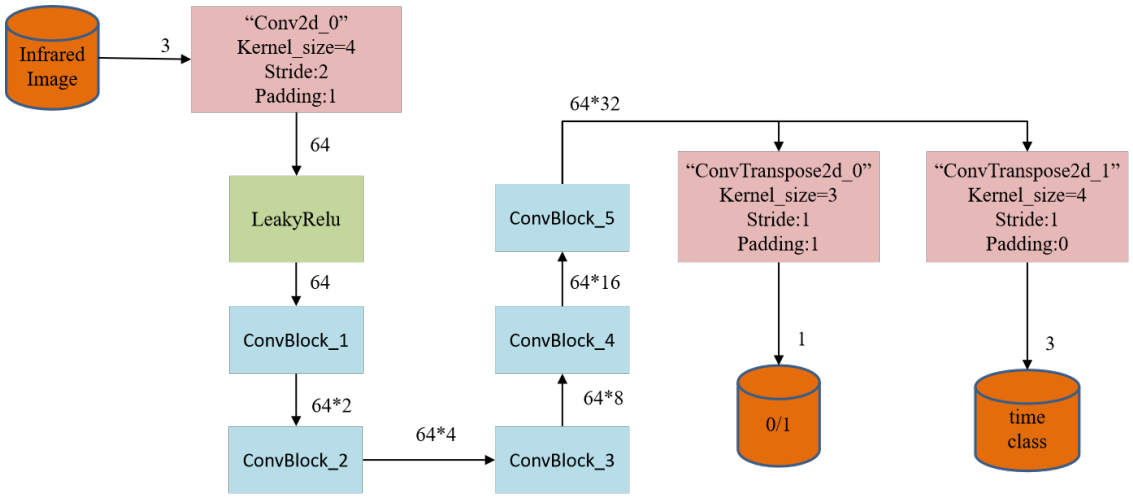


Fig. 9: StarGAN Discriminator Network Architecture

Module	**Convolution Block Name**	**Convolution Layer Parameters** (f,k,s,p)	**Input Features** (b,c,h,w)	**Output Features** (b,c,h,w)
Input	-	-	(1,3,256,256)	(1,6,256,256)
Encoder	CB0	(64,7,3,1)	(1,6,256,256)	(1,64,256,256)
	CB1	(128,4,1,2)	(1,64,256,256)	(1,128,128,128)
	CB2	(256,4,1,2)	(1,128,128,128)	(1,256,64,64)
Transmission Layer	RCB0	(256,3,1,1)	(1,256,64,64)	(1,256,64,64)
		(256,3,1,1)	(1,256,64,64)	(1,256,64,64)
	RCB1	(256,3,1,1)	(1,256,64,64)	(1,256,64,64)
		(256,3,1,1)	(1,256,64,64)	(1,256,64,64)
	RCB2	(256,3,1,1)	(1,256,64,64)	(1,256,64,64)
		(256,3,1,1)	(1,256,64,64)	(1,256,64,64)
	RCB3	(256,3,1,1)	(1,256,64,64)	(1,256,64,64)
		(256,3,1,1)	(1,256,64,64)	(1,256,64,64)
	RCB4	(256,3,1,1)	(1,256,64,64)	(1,256,64,64)
		(256,3,1,1)	(1,256,64,64)	(1,256,64,64)
	RCB5	(256,3,1,1)	(1,256,64,64)	(1,256,64,64)
		(256,3,1,1)	(1,256,64,64)	(1,256,64,64)
Decoder	DB0	(128,4,1,2)	(1,256,64,64)	(1,128,128,128)
	DB1	(64,4,1,2)	(1,128,128,128)	(1,64,256,256)
	DB2	(3,7,1,1)	(1,64,256,256)	(1,3,256,256)

Table 1: StarGAN Generator Network Parameters

The parameters of the D-network are shown in Table 2. The convolution layer parameters (f, k, s, p) represent the number of filters, filter size, stride, and padding size, respectively. The input and output feature parameters (b, c, h, w) represent batch size, number of channels, feature map height, and feature map width, respectively. The output module includes two branches: the time period information classification branch and the real/fake sample judgment branch.

3.1.3 StarGAN Loss Functions

Generative adversarial networks (GANs) are powerful models for data generation but suffer from training instability, vanishing gradients, and exploding gradients. This instability arises because the generator's optimization goal under the optimal discriminator minimizes the JS

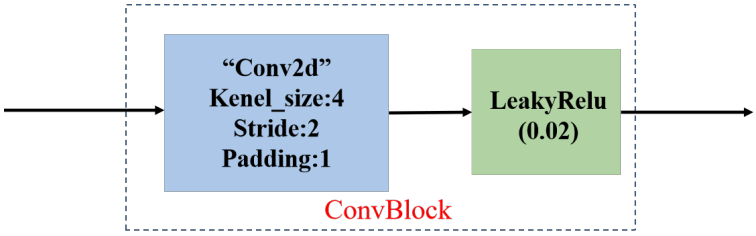


Fig. 10: ConvBlock Submodule Network Architecture

Module	Convolution Block Name	Convolution Parameters (f, k, s, p)	Input Features (b, c, h, w)	Output Features (b, c, h, w)
Input	-	-	$(1, 3, 256, 256)$	-
Downsampling	Conv2d_0	$(64, 4, 1, 2)$	$(1, 3, 256, 256)$	$(1, 64, 128, 128)$
	ConvBlock_1	$(128, 4, 1, 2)$	$(1, 64, 128, 128)$	$(1, 128, 64, 64)$
	ConvBlock_2	$(256, 4, 1, 2)$	$(1, 128, 64, 64)$	$(1, 256, 32, 32)$
	ConvBlock_3	$(512, 4, 1, 2)$	$(1, 256, 32, 32)$	$(1, 512, 16, 16)$
	ConvBlock_4	$(1024, 4, 1, 2)$	$(1, 512, 16, 16)$	$(1, 1024, 8, 8)$
	ConvBlock_5	$(2048, 4, 1, 2)$	$(1, 1024, 8, 8)$	$(1, 2048, 4, 4)$
Output Module	Contranspose2d_0	$(1, 3, 1, 1)$	$(1, 2048, 4, 4)$	$(1, 1, 4, 4)$
	Contranspose2d_1	$(3, 4, 0, 1)$	$(1, 2048, 4, 4)$	$(1, 3, 1, 1)$

Table 2: Network structure of the StarGAN discriminator.

divergence between the real and generated distributions. JS divergence effectively measures the similarity between two distributions but fails when their support regions are disjoint—a common scenario in high-dimensional data mappings.

To address these issues, the Wasserstein distance was introduced, which remains effective even when the two distributions do not overlap. The Wasserstein loss in WGAN is defined as follows:

$$W(P_1, P_2) = \sup_{\|f\|_L \leq K} \mathbb{E}_{x \sim P_1}[f(x)] - \mathbb{E}_{x \sim P_2}[f(x)] \tag{4}$$

where P_1 and P_2 are the real and generated data distributions, and f is a Lipschitz-continuous function.

In WGAN, weight clipping is used to enforce the Lipschitz constraint, restricting discriminator parameters $w_i \in [-0.01, 0.01]$. However, weight clipping can lead to slow convergence and parameter saturation, which reduces the discriminator's capacity. Gradient clipping

thresholds that are too high or low can further cause gradient explosion or vanishing, complicating parameter tuning.

The WGAN-GP model replaces weight clipping with a gradient penalty. An additional loss term penalizes deviations from the Lipschitz condition:

$$W(P_1, P_2) = \mathbb{E}_{x\sim P_1}[f(x)] - \mathbb{E}_{x\sim P_2}[f(x)] - \lambda \mathbb{E}_{\hat{x}\sim P_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \tag{5}$$

where $\hat{x} = \epsilon x_1 + (1-\epsilon)x_2$, $x_1 \in P_1$, $x_2 \in P_2$, and $\epsilon \sim \text{Uniform}[0,1]$.

WGAN-GP avoids training difficulties and stabilizes the training process. Therefore, this work adopts WGAN-GP as the adversarial loss for unpaired image generation. The adversarial loss for the generator (G) and discriminator (D) is defined as:

$$L_{adv} = \mathbb{E}_x[D_{src}(x)] - \mathbb{E}_{(x,c)}[D_{src}(G(x,c))] - \lambda_{gp}\mathbb{E}_{\hat{x}}[(\|\nabla_{\hat{x}} D_{src}(\hat{x})\|_2 - 1)^2] \tag{6}$$

Temporal Classification Loss For a given input infrared image x and target domain label c, temporal expansion transforms x into an output image with the desired temporal features. To achieve this, D includes an auxiliary classifier, adding temporal classification losses for both real (L^r_{cls}) and generated (L^f_{cls}) images:

$$L^r_{cls} = \mathbb{E}_{(x,c')}\left[-c' \log(D_{cls}(x))\right] \tag{7}$$

$$L^f_{cls} = \mathbb{E}_{(x,c)}\left[-c \log(D_{cls}(G(x,c)))\right] \tag{8}$$

where c' is the original label and D_{cls} represents the discriminator's classification branch.

Reconstruction Loss To ensure generated images preserve the input scene content, a cycle consistency loss is applied:

$$L_{rec} = \mathbb{E}_{(x,c,c')}[\|x - G(G(x,c),c')\|_1] \tag{9}$$

Optimization Objectives The overall loss functions for G and D are:

$$L_G = L_{adv} + \lambda_{cls} L^f_{cls} + \lambda_{rec} L_{rec} \tag{10}$$

$$L_D = L_{adv} + \lambda_{cls} L^r_{cls} \tag{11}$$

where λ_{cls} and λ_{rec} control the relative importance of classification and reconstruction losses.

Figure 11 illustrates the overall structure and relationships of these loss functions.

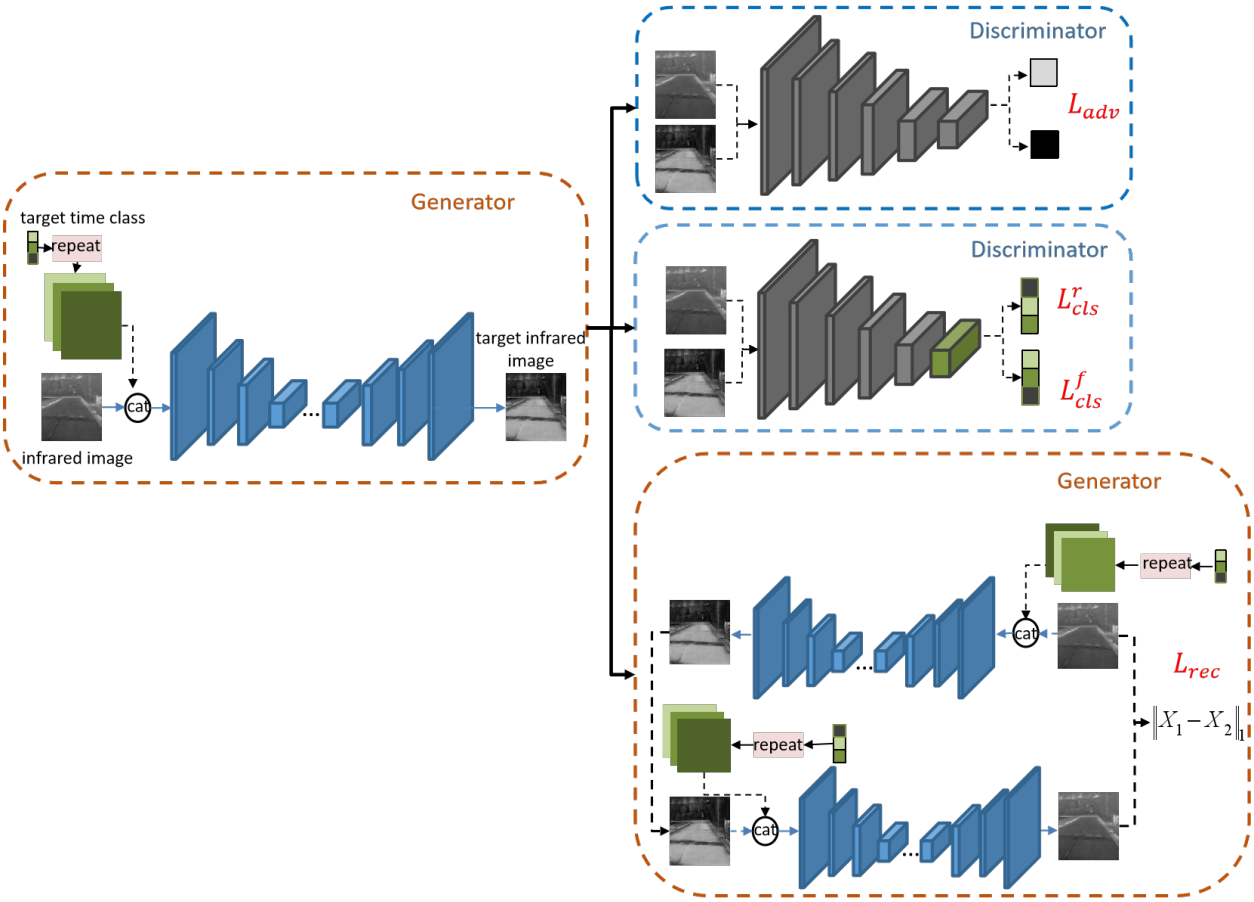


Fig. 11: StarGAN Loss Function Diagram.

3.1.4 Model Training and Experimental Results Analysis

Dataset Description The dataset used in the experiments is the road scene dataset released by the Kaist Laboratory. Infrared images in the dataset were captured by a FLIR infrared camera with a wavelength range of 7.5 μm to 13 μm . This study focuses on infrared images captured at three specific time periods: 5 a.m., 2 p.m., and 7 p.m. For each time period, the training dataset consists of 2000 images, and the testing dataset consists of 200 images. It is worth noting that the infrared images are not paired across time periods; however, the scenes in the dataset are similar, mainly including roads, trees, grass, vehicles, and a small number of pedestrians.

Training Details and Parameter Settings To preprocess the dataset, all images

were cropped and resized to a size of (256, 256). Each infrared image was labeled with its corresponding time period, and the label was combined with the infrared image to form a data sample $\{0 : \text{infrared image}, 1 : \text{label}\}$. The same processing was applied to all time periods, resulting in a total of 6000 training data samples.

The training parameters were configured as follows: a batch size of 8 was used, with a total of 20,000 iterations. The learning rate was initially fixed at 0.0001 for the first 10,000 iterations, after which it was decreased by 0.00001 every 1,000 iterations until reaching 0. The Adam optimizer was employed for optimization, with $\beta_1 = 0.5$ and $\beta_2 = 0.999$.

Time period information was encoded using one-hot encoding. The one-hot encoding was expanded to a shape of (256, 256) and concatenated with the infrared image along the second dimension as input to the generator network.

Training Process and Results Analysis The network was trained on the infrared time period dataset using the designed loss functions. After 100 epochs, the trends of the generator and discriminator losses are shown in Figures 12 and 13.

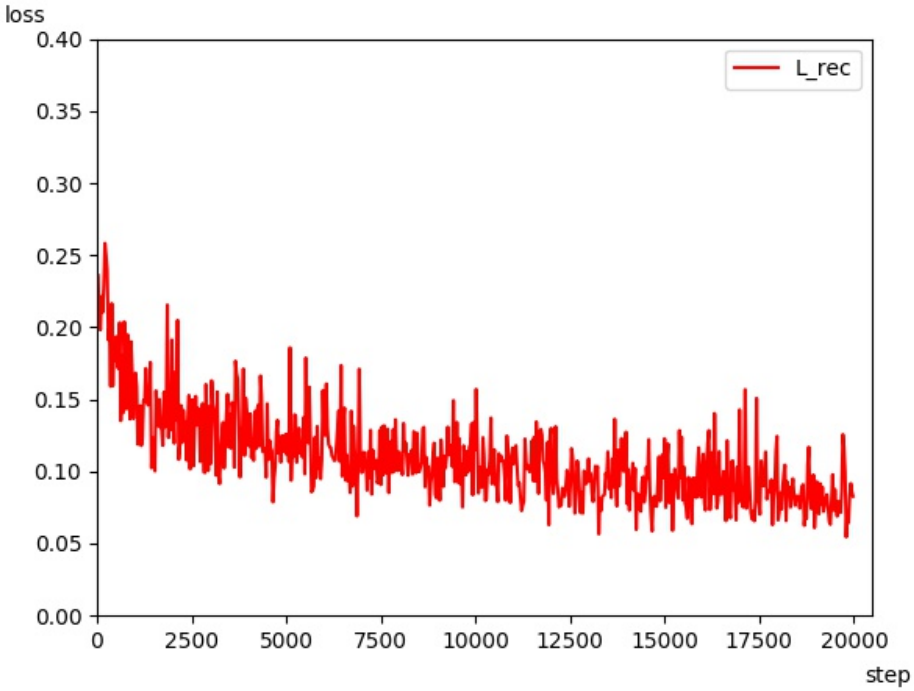


Fig. 12: StarGAN Generator Loss Curve

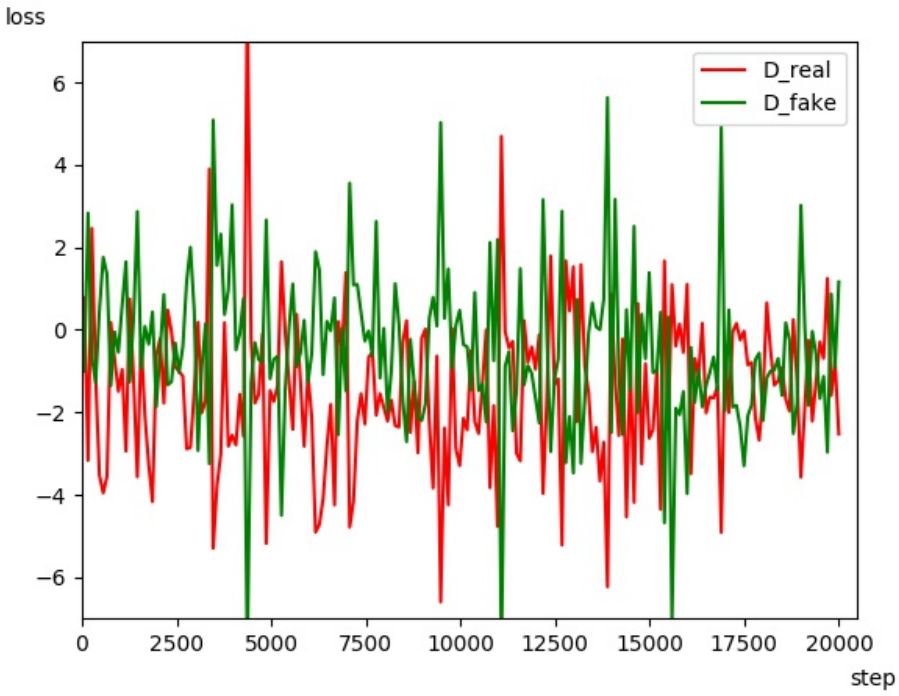


Fig. 13: StarGAN Discriminator Loss Curve

The results show that the cycle consistency loss steadily decreases and gradually converges after 17500 steps. The discriminator's oscillations diminish over time, and the discrimination loss for real and fake samples becomes comparable, indicating no mode collapse.

Experimental Results Using the trained generator network, the temporal extension of the test set data was performed. In this section, the subjective evaluation method is first adopted to measure the realism of the extended images and the effectiveness of the temporal information. Figures 14 shows the experimental results of infrared images at 5 a.m. being extended to 2 p.m. and 7 p.m. The first column contains the input infrared images at 5 a.m., while the second, third, and fourth columns show the infrared images at 5 a.m., 2 p.m., and 7 p.m., respectively, after temporal extension.

In Figures 14, the three scenes in the first column consist of real infrared images captured from roads, buildings, and trees. It can be observed that the extension results for 5 a.m. and 7 p.m. are better, as no abrupt changes in brightness occur within the semantic range of each scene. However, the extension results for 2 p.m. show varying degrees of distortion. Examining the road semantic scene, the results in the first row are satisfactory, but the

second and third rows fail to extend the road information in the distant regions, retaining the grayscale characteristics of the original infrared image at 5 a.m., which appear darker. The analysis indicates that the grayscale average of infrared images at 5 a.m. is relatively low, with low contrast. Infrared images at 7 p.m. generally exhibit higher brightness, with imaging effects relatively closer to 5 a.m., resulting in better extension results. In contrast, the 2 p.m. infrared images not only have higher brightness but also exhibit significant changes in grayscale information for different semantic scenes, with greater contrast, leading to poorer generation results.

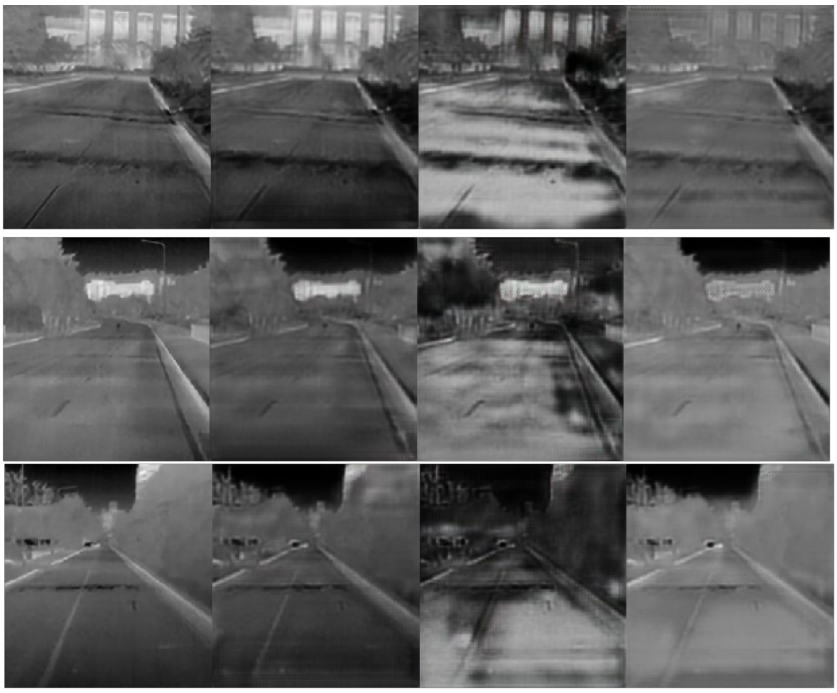


Fig. 14: Temporal extension results for infrared images captured at 5 a.m.

Figures 15 presents the temporal extension results for infrared images captured at 2 p.m. Compared to 5 a.m., the imaging effect is significantly improved. The analysis suggests that the enhanced performance is primarily due to the higher contrast of 2 p.m. images, which provides clearer contours between different materials.

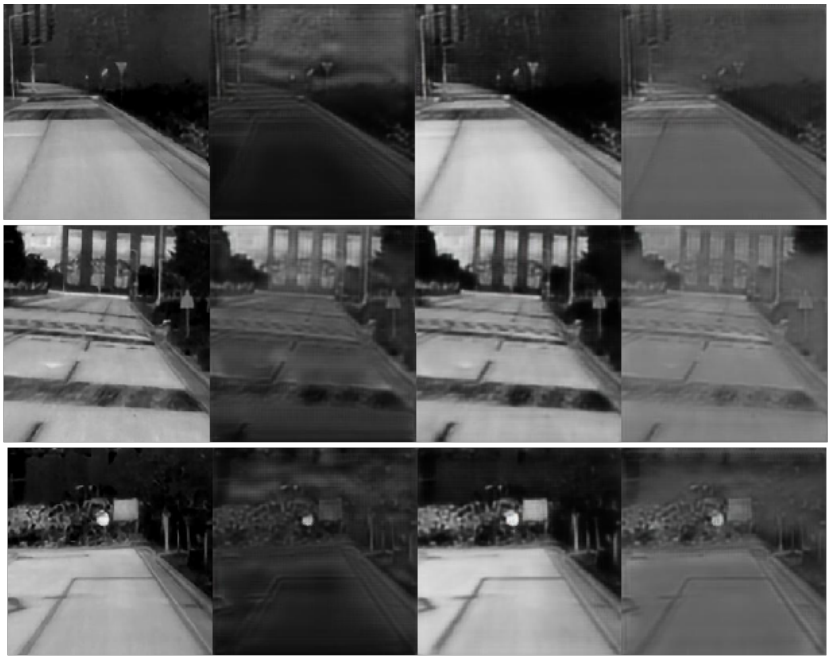


Fig. 15: Temporal extension results for infrared images captured at 2 p.m.

Figures 16 illustrates the temporal extension results for infrared images captured at 7 p.m. Similar issues with poor image quality arise due to weaker contrast. Additionally, it can be observed that extending from 7 p.m. to 5 a.m. results in errors in the tree semantic scene. Trees, having a smaller heat capacity and less variation in infrared radiation, generally exhibit darker grayscale values relative to roads. However, in the first and second rows of the figure, the grayscale of trees in the extended 5 a.m. infrared images becomes noticeably brighter.

From the above analysis, it can be concluded that the StarGAN network initially achieves the task of temporal extension, producing satisfactory results under certain scenarios. However, issues such as poor image quality and errors in temporal extension of infrared textures are also observed. These issues are partly attributed to the inherent characteristics of infrared imaging, such as low grayscale and contrast. Additionally, the dataset used in this study includes significant variations in scene information, leading to notable semantic scene

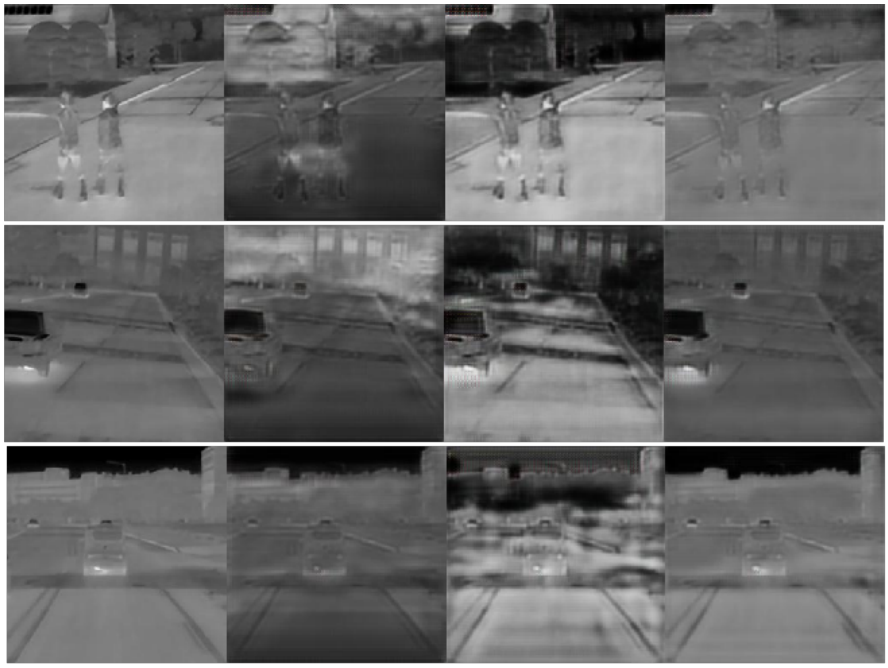


Fig. 16: Temporal extension results for infrared images captured at 7 p.m.

generation errors in some results. To preserve the original scene information while ensuring strong generalization capabilities of the network, this paper proposes a semantically constrained StarGAN network.

3.2 Infrared Image Temporal Extension Based on Semantically Constrained StarGAN

3.2.1 Semantically Constrained StarGAN Network

In the task of infrared image temporal extension, the algorithm must satisfy two essential requirements: (1) preserving the semantic information of the original scene in the generated infrared image and (2) ensuring that the generated infrared image accurately reflects the temporal texture distribution characteristics of the corresponding time period. To address

these requirements, this paper introduces a semantically constrained StarGAN with key enhancements to guarantee semantic invariance and temporal information accuracy.

1. **Semantic Invariance:** Expanding on the original StarGAN architecture, we incorporate a semantic information encoding branch within the generator. Furthermore, a semantic encoding consistency loss function is integrated into the training process. These additions ensure that the generated infrared image retains the semantic features of the original scene, maintaining alignment of semantic information across different time periods.

2. **Temporal Information Accuracy:** Temporal variations in infrared radiation differ across objects due to their physical properties. For instance, objects with high specific heat capacity exhibit greater day-night temperature fluctuations and more pronounced radiation changes, while those with low specific heat capacity demonstrate smoother temperature transitions and subtler radiation variations. To capture these distinctions, we propose a temporal information encoding method that leverages semantic segmentation maps and temporal infrared radiation curves. This enables the generator to model time-specific radiation characteristics with high precision.

The architecture of the semantically constrained StarGAN for infrared image temporal extension is illustrated in Figure 17. The generator's encoder comprises three branches: a temporal information encoding branch, a semantic encoding branch, and an image feature encoding branch. Outputs from these branches are concatenated and fed into the decoder, which generates infrared images aligned with the temporal encoding data. The discriminator network retains the original structure of the StarGAN architecture.

By addressing the dual challenges of semantic invariance and temporal information accuracy, this enhanced StarGAN framework offers a robust solution for infrared image temporal extension. It holds significant potential for applications such as remote sensing, surveillance, and thermal imaging, where accurate temporal extension of infrared imagery is critical.

The encoder of the generator (G) network builds upon the StarGAN architecture, integrating additional temporal encoding and semantic encoding branches. The temporal information encoding branch calculates the temporal code using semantic segmentation results from the semantic encoding branch and the temporal curve of infrared radiation, as described in Section 3.2.2. The semantic encoding branch is a Unet-based semantic segmentation network that performs pixel-level classification of input infrared images. During training, supervised learning with labeled data is employed to train the semantic segmentation branch.

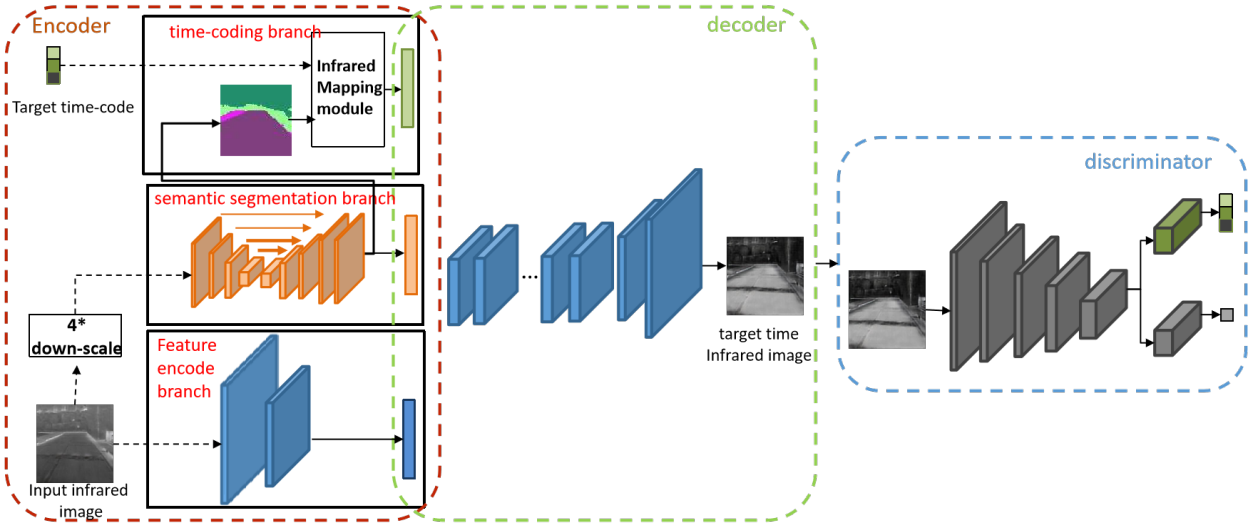


Fig. 17: Semantically Constrained StarGAN Model Structure

If the semantic segmentation branch accurately captures the semantic information, the penultimate feature layer of this network encapsulates the semantic information of the input image. Consequently, this feature layer is concatenated with the encoder output as temporal-independent features, preserving the original infrared image details while ensuring semantic invariance during temporal extension. To enforce consistency in semantic information, a semantic encoding consistency loss is added during training, ensuring that the extended infrared image retains the semantic properties of the original. Further details of this implementation are provided in Section 3.2.3.

The semantically constrained StarGAN for infrared image temporal extension follows the algorithmic flow outlined below:

1. **Train the D Network**: The real infrared images and the infrared images generated by the generator are input into the discriminator (D) network. The D network learns to distinguish between real and generated infrared images and classifies them into their respective temporal domains.

2. **Train the G Network (Source to Target Mapping)**: Infrared images are input into the semantic encoding branch and the image feature encoding branch. A random target temporal label is generated. Using this target temporal label and the segmentation results from the semantic encoding branch, the temporal encoding $label_r$ is computed. The outputs of the image feature encoding branch, the penultimate feature

layer of the semantic encoding branch, and the temporal encoding $label_r$ are concatenated and passed to the decoder, which generates the infrared image corresponding to the target temporal encoding.

3. **Train the G Network (Target to Source Mapping)**: The generator (G) network reconstructs the original infrared image from the generated image, given the original temporal encoding label. The generated image is input into the semantic encoding branch and the image feature encoding branch. Using the original temporal label and the segmentation results of the semantic encoding branch, the temporal encoding $label_s$ is computed. The outputs of the image feature encoding branch, the penultimate feature layer of the semantic encoding branch, and the temporal encoding $label_s$ are concatenated and passed to the decoder to reconstruct the original infrared image $\hat{x}^{rec}$.

4. **Train the D Network ("Fooling" the Discriminator)**: The infrared images generated by the G network are input into the D network. The discriminator evaluates the authenticity of the images and outputs the corresponding temporal encoding information. At this stage, the generator aims to "fool" the discriminator by making it classify the generated images as real and correctly assign them to their respective temporal domains.

3.2.2 Semantic-Constrained StarGAN Time Period Encoding Method

In the StarGAN network, the time period encoding information among different materials in full-frame infrared images is identical. However, infrared radiation is closely related to material properties, and the infrared radiation distribution of different materials conforms to their respective diurnal variation characteristics. Based on this, utilizing prior knowledge of the diurnal variation characteristics of infrared radiation for different materials, this paper proposes a time period encoding method based on semantic segmentation maps and infrared radiation time-varying curves.

The variation in radiation for different materials is closely related to temperature changes. We can construct a database of diurnal infrared radiation variations for different materials as prior knowledge for infrared image time period extension. To statistically analyze the time-varying curves of infrared radiation for different materials, this paper uses the SE-WORKBENCH simulation software to collect infrared radiation data for several materials. SE-WORKBENCH is a widely used infrared simulation software. By segmenting scene materials and using pre-calculated atmospheric parameters, it can simulate infrared images of any scene in different spectral bands and time periods.

In the dataset used in this paper, the scenes are similar and mainly include roads, trees, grass, vehicles, and a small number of pedestrians. Using SE-WORKBENCH, we collected time-varying infrared radiation data for the four frequently appearing scenes of sky, grass, ground, and trees in the dataset, as shown in Table 3. According to the table, the variation in sky radiation remains relatively stable throughout the day; the radiation trends and amplitudes for grass and trees are almost identical because vegetation has a low specific heat capacity, resulting in smaller temperature and infrared radiation variations; the ground has a higher specific heat capacity, leading to relatively larger infrared radiation variations.

Table 3: Statistics of Radiation Variation for Different Materials Over Time (Unit: $W/(m^2 \cdot sr)$)

Time	Sky	Tree	Grass	Ground
05:00	1.48125	1.53412	1.52892	1.57349
14:00	1.55423	2.39011	2.53156	3.28729
19:00	1.46777	1.57837	1.59181	1.63271

For material segmentation in infrared images, the semantic segmentation results from Section 4.2.1 are used as the corresponding material segmentation results. This equivalence holds because this paper only maps materials to infrared radiation for the four scenarios of sky, grass, ground, and trees. Within these scenarios, the materials are uniform, with no ambiguous information. Based on the constructed database of diurnal infrared radiation variations for different materials and semantic segmentation results, this paper proposes a time period encoding method based on semantic segmentation maps and infrared texture time-varying curves. The implementation is as follows:

1. First, normalize the collected mid-wave infrared radiation variations. The normalized results are shown in Table 4.

2. Based on the semantic segmentation map from the original infrared image, combine the semantic encoding information of the scene with the target time period information to find the corresponding normalized radiation value in Table 4 and replace the pixel values on the semantic map.

3. Repeat the above operation for each pixel on the semantic map. For pixels corresponding to semantic encodings other than sky, grass, ground, and trees, set the value to 0 to obtain material-related time period encoding information.

4. To ensure the feasibility of time period extension for materials other than sky, grass, ground, and trees, merge the material-related time period encoding information with the time period encoding information proposed in Section 4.1.1 at the channel level to serve as the time period encoding for the semantic-constrained StarGAN.

Table 4: Mapping Table for Time Period Encoding Information of Different Materials

Time	Sky	Tree	Grass	Ground
05:00	0.007	0.036	0.034	0.058
14:00	0.047	0.507	0.585	1
19:00	0.000	0.061	0.068	0.091

3.2.3 Semantic-Constrained StarGAN Model Construction and Analysis

For the task of infrared image extension, this paper proposes a semantic-constrained StarGAN based on the StarGAN network. A semantic encoding branch is added, and the features from the second-to-last layer of the semantic encoding branch are used as the semantic encoding of the infrared image. Additionally, a time period encoding method based on the time-varying priors of infrared radiation for different materials is proposed. The discriminator network structure is identical to the one introduced in Section 3.1.2. This subsection focuses on the generator network structure.

The generator network structure constructed in this paper is shown in Figure 18. The semantic-constrained StarGAN network includes two main components: an encoder and a decoder. The encoder consists of three encoding branches: the time period encoding branch, the semantic encoding branch, and the image feature encoding branch. In the figure, the red dashed box represents the encoder, and the green dashed box represents the decoder. The image feature encoding branch is the same as the encoder in the StarGAN network described in Section 3.1.2. The decoder is a combination of the transmission layer decoder from the StarGAN network. The semantic encoding branch is based on a U-net structure, and the time period encoding branch maps the semantic segmentation results to infrared radiation values.

Except for the semantic encoding branch, the convolutional block structures in the generator network are identical to those in Section 3.1.2 and will not be analyzed further here. The semantic encoding module primarily consists of an input convolutional layer, four encoder submodules, four decoder submodules, and an output convolutional layer. The structures of the encoder and decoder submodules are shown in Figures 19 and 20, respectively.

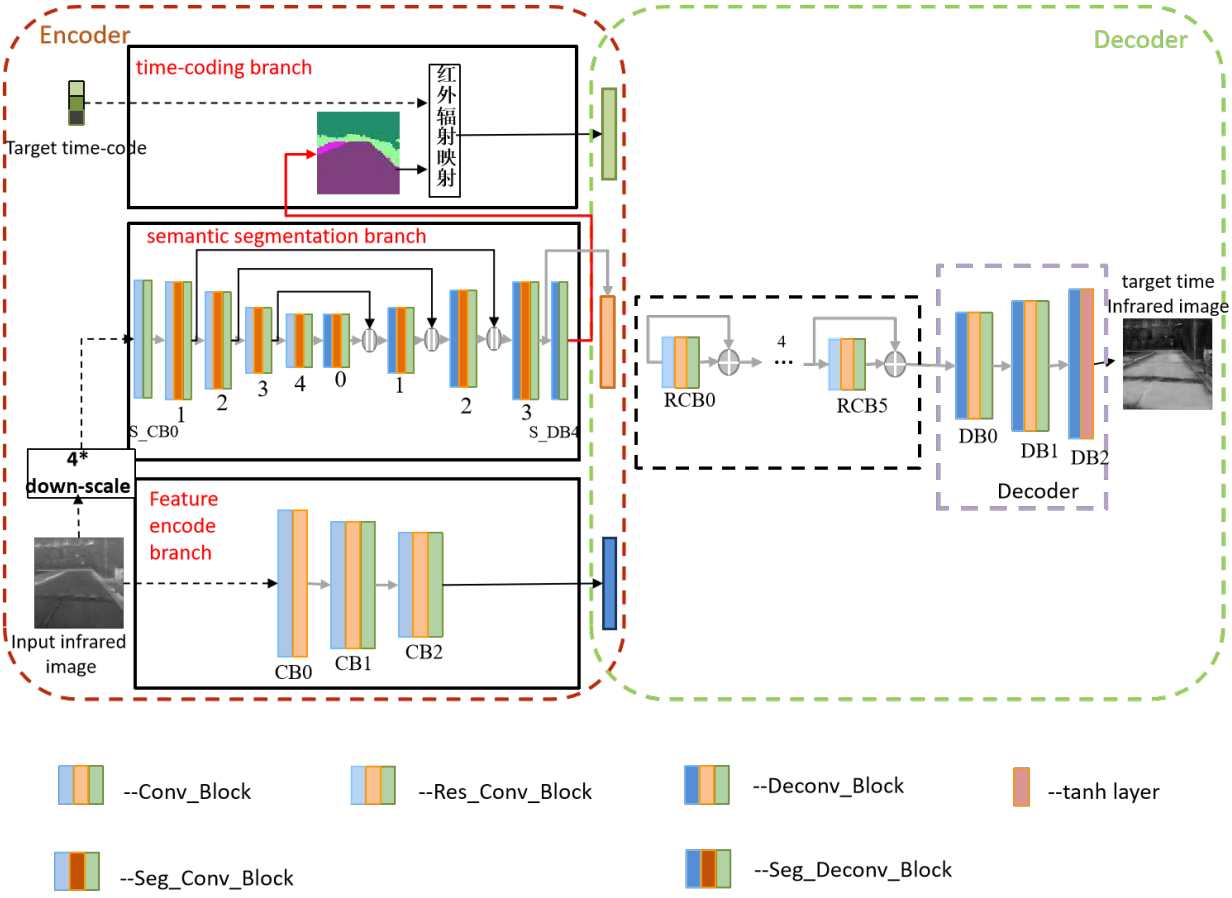


Fig. 18: Semantic-Constrained StarGAN Generator Network Structure

3.2.4 Generator Network Parameters of Semantic-Constrained StarGAN

The parameters of the G network are shown in Table 5. In the table, the convolutional layer parameters (f, k, s, p) represent the number of filters, filter size, stride, and padding, respectively. The input and output parameters (b, c, h, w) represent the batch size, number of channels, height, and width of the feature map, respectively. The parameters for the image feature encoding branch are the same as those in the StarGAN encoder. The decoder structure is identical to the transmission layer and decoder in the StarGAN network, but the parameter settings are different. In Table 5, the RCBs in the decoder module represent the six submodules in the transmission layer, where each submodule has identical parameters.

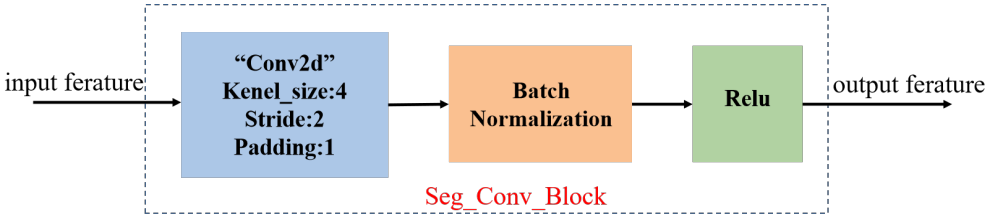


Fig. 19: Network Structure of the Encoder Submodule in the Semantic Encoding Branch

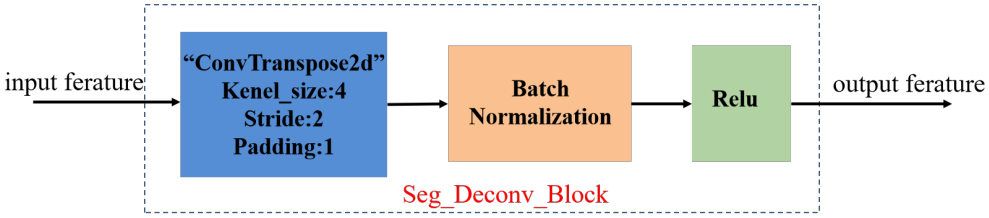


Fig. 20: Network Structure of the Decoder Submodule in the Semantic Encoding Branch

3.2.5 Semantic-Constrained StarGAN Loss Function

Based on the StarGAN loss function introduced in Section 3.1.3, the Semantic-Constrained StarGAN network adds two loss functions: the semantic segmentation loss function and the semantic encoding consistency loss function.

Semantic Segmentation Loss Function For training the semantic segmentation branch, this paper adopts a supervised learning approach with labeled data and uses the cross-entropy loss function for network training. The calculation is as follows:

$$L_{\text{seg}}^{r} = \mathbb{E}_{(x,c^s)}\left[-c^s \log(S(x))\right] \tag{12}$$

where c^s represents the scene category, and $S(x)$ represents the semantic segmentation output.

Semantic Encoding Consistency Loss Function To ensure that the semantic information in the generated infrared image is consistent with that of the input infrared image, this paper introduces the semantic encoding consistency loss. The generated infrared image, after passing through the semantic encoding branch, should output semantic segmentation results that match the original infrared image's semantic information, as shown in the following equation:

$$L_{\text{rec\_seg}} = \mathbb{E}_{(x,c,c')}\left[-c^s \log(S(G(x,c)))\right] \tag{13}$$

Table 5: Semantic-Constrained StarGAN Generator Network Parameters

Module	Convolutional Block Name	Convolutional Layer Parameters (f, k, s, p)	Input Features (b, c, h, w)	Output Features (b, c, h, w)
Semantic Encoding Module	S_CB0	(64, 4, 1, 1)	(1, 3, 64, 64)	(1, 64, 64, 64)
	S_CB1	(128, 4, 1, 2)	(1, 64, 64, 64)	(1, 128, 32, 32)
	S_CB2	(256, 4, 1, 2)	(1, 128, 32, 32)	(1, 256, 16, 16)
	S_CB3	(512, 4, 1, 2)	(1, 256, 16, 16)	(1, 512, 8, 8)
	S_CB4	(1024, 4, 1, 2)	(1, 512, 8, 8)	(1, 1024, 4, 4)
	S_DB0	(512, 4, 1, 2)	(1, 1024, 4, 4)	(1, 512, 8, 8)
	S_DB1	(256, 4, 1, 2)	(1, 512*2, 8, 8)	(1, 256, 16, 16)
	S_DB2	(128, 4, 1, 2)	(1,256*2,16,16)	(1, 128, 32, 32)
	S_DB3	(64, 4, 1, 2)	(1,128*2, 32, 32)	(1, 64, 64, 64)
	S_DB4	(20, 4, 1, 1)	(1, 64, 64, 64)	(1, 20, 64, 64)
Decoder	RCBs	(336, 3, 1, 1)	(1, 336, 64, 64)	(1, 336, 64, 64)
	DB0	(128, 4, 1, 2)	(1, 336, 64, 64)	(1,128, 128, 128)
	DB1	(64, 4, 1, 2)	(1,128, 128, 128)	(1, 64, 256, 256)
	DB2	(3, 7, 1, 1)	(1, 64, 256, 256)	(1, 3, 256, 256)

Generator and Discriminator Optimization Loss Functions The optimization loss functions for the generator and discriminator are as follows:

$$L_D = -L_{\text{adv}} + \lambda_{\text{cls}} L^r_{\text{cls}} \tag{14}$$

$$L_G = L_{\text{adv}} + \lambda_{\text{cls}} L^f_{\text{cls}} + \lambda_{\text{rec}} L_{\text{rec}} + \lambda_{\text{seg}} L^r_{\text{seg}} + \lambda_{\text{rec\_seg}} L_{\text{rec\_seg}} \tag{15}$$

where λ_{cls}, λ_{rec}, λ_{seg}, and $\lambda_{\text{rec\_seg}}$ are hyperparameters that control the relative importance of the time period classification loss, reconstruction loss, semantic segmentation loss, and semantic encoding consistency loss relative to the adversarial loss.

The diagram for the loss function of infrared image time period extension based on Semantic-Constrained StarGAN is shown in Figures 21.

3.2.6 Evaluation Metrics for Infrared Image Time Period Extension

Grayscale Distribution Histogram In this paper, we use the grayscale distribution histogram to estimate the grayscale distribution of images. The formula for calculating the grayscale distribution histogram is as follows:

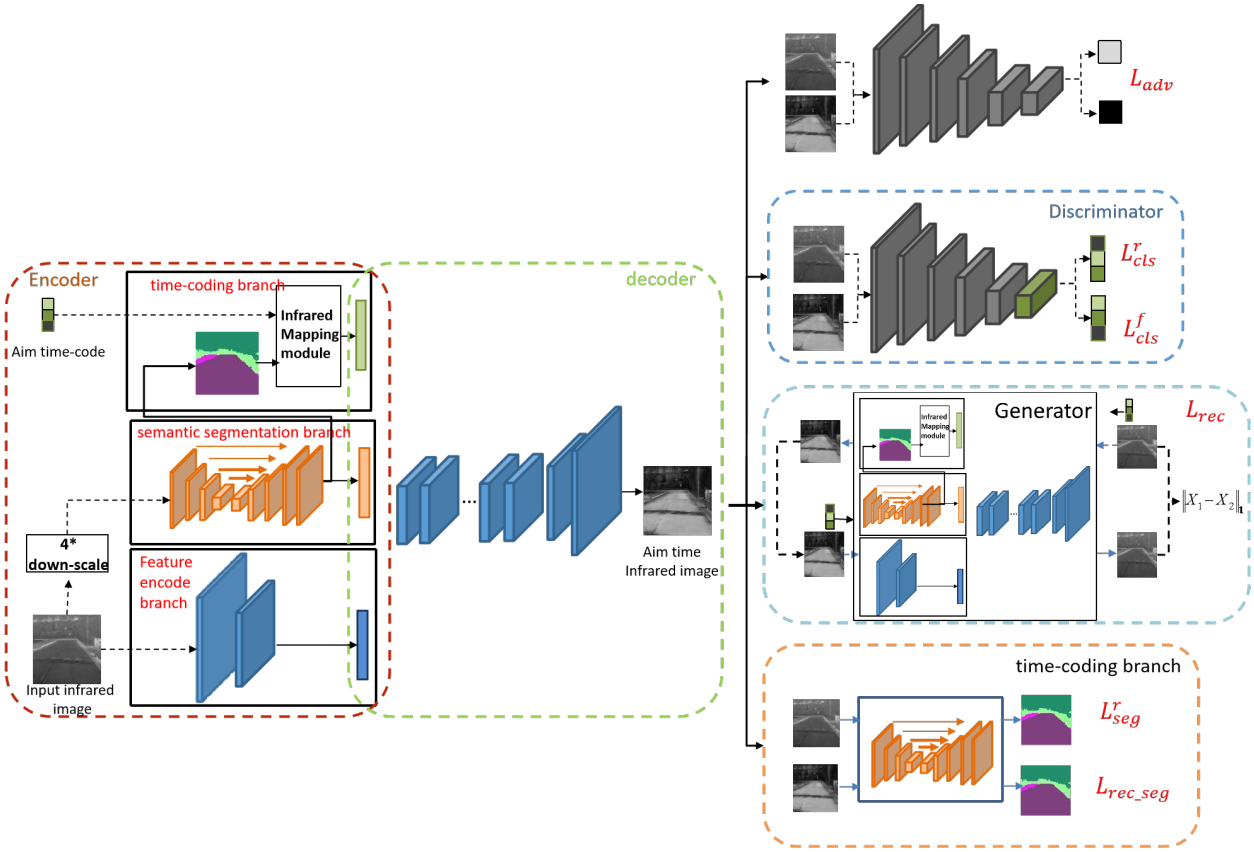


Fig. 21: Semantic-Constrained StarGAN Loss Function Diagram

$$p_h = \frac{n_h}{\sum_h n_h} \tag{16}$$

where h represents the grayscale value (0-255), p_h represents the probability of the grayscale value h appearing in the image, and n_h is the number of pixels with grayscale value h.

Bhattacharyya Coefficient The Bhattacharyya coefficient is used to measure the consistency of histogram distributions. Its calculation is as follows:

$$BC(p,q) = \sum_{i=1}^{n} \sqrt{p_i q_i} \tag{17}$$

where p and q are the histograms of the real infrared image and the extended infrared

image, and n is the number of histogram bins.

Structural Similarity Since the dataset used in this paper does not have corresponding ground truth labels for images, to evaluate the invariance of the image structure and corresponding semantic structure after time period extension, we use SSIM (Structural Similarity Index) to measure the structural similarity between two infrared images before and after the time period extension. SSIM evaluates the structural information of two images, and the higher the value, the greater the structural consistency between the two images.

3.2.7 Model Algorithm and Experimental Results Analysis

Based on the Semantic-Constrained StarGAN network constructed in Section 3.2.3, experiments were conducted using the same dataset as in Section 3.1.4. After 20,000 iterations, the network converged, and the generator's cycle consistency loss and the discriminator's loss function are shown in Figures 22 and Figures 23, respectively. The generator's loss function continuously decreased, and the discriminator's ability to distinguish between real and fake samples remained relatively consistent, with no gradient collapse observed. Compared to the training process of the StarGAN network, the oscillation in the discriminator's loss function was significantly reduced.

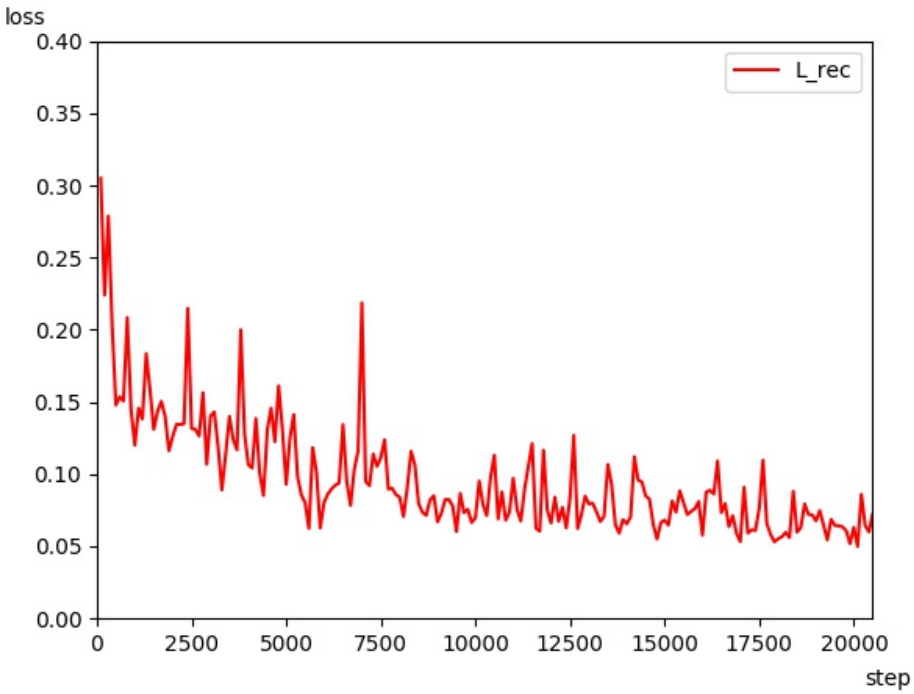


Fig. 22: Semantic-Constrained StarGAN Generator Cycle Consistency Loss Curve

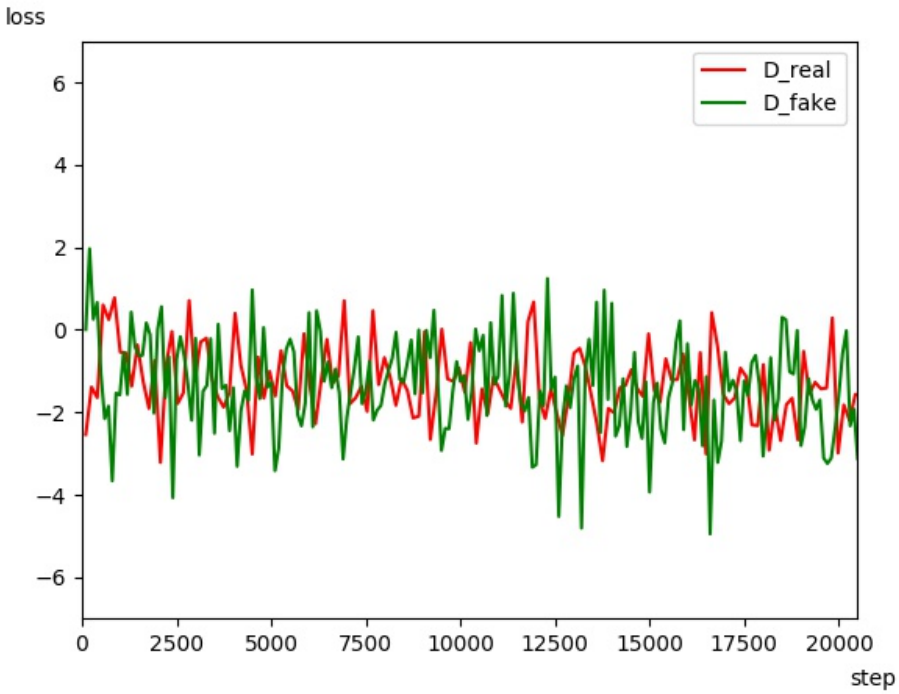
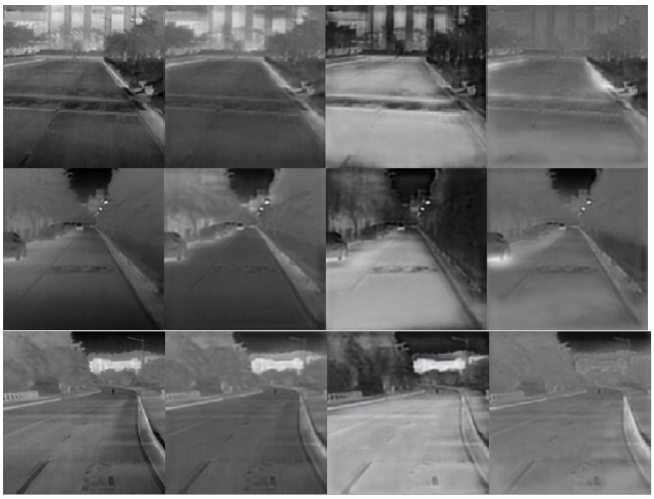


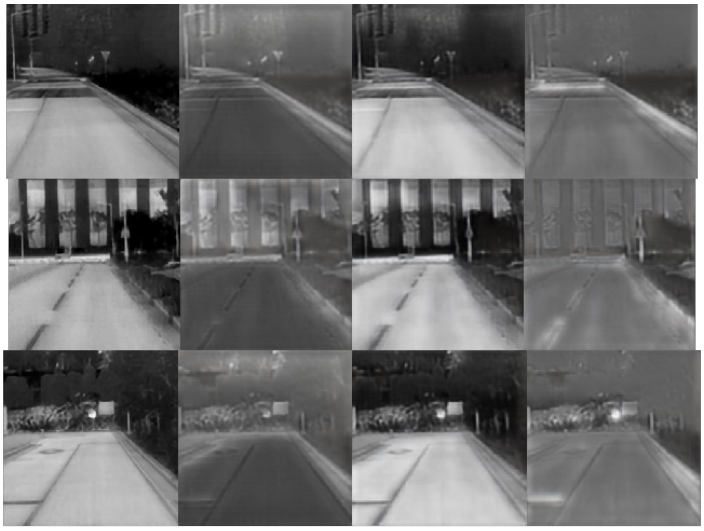
Fig. 23: Semantic-Constrained StarGAN Discriminator Loss Curve

The trained generator network was used for time period extension on the test set. This section first uses subjective judgment to assess the realism of the generated images and the effectiveness of the time period information. Figures 24, Figures 25, and Figures 26 show the time period extension results for 5:00 AM, 2:00 PM, and 7:00 PM, respectively. The first column shows the input real infrared images (the same scenes used in Section 3.1.4), and the second, third, and fourth columns show the infrared images generated for 5:00 AM, 2:00 PM, and 7:00 PM, respectively.



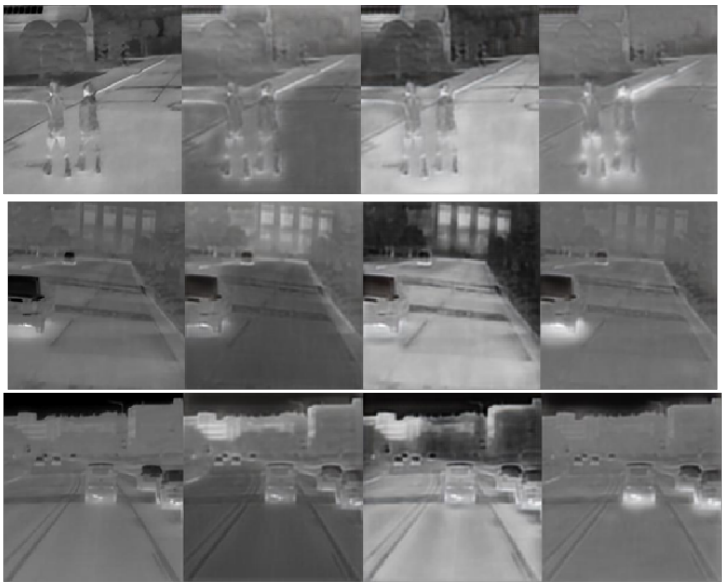
Real Image: Night (5 AM) & Synthetic Image: 5 AM & Synthetic Image: 2 PM & Synthetic Image: 7 PM

Fig. 24: 5:00 AM Infrared Image Time Period Extension Results



Real Image: Night (2 PM) & Synthetic Image: 5 AM & Synthetic Image: 2 PM & Synthetic Image: 7 PM

Fig. 25: 2:00 PM Infrared Image Time Period Extension Results



Real Image: Night (7 PM) & Synthetic Image: 5 AM & Synthetic Image: 2 PM & Synthetic Image: 7 PM

Fig. 26: 7:00 PM Infrared Image Time Period Extension Results

From Figures 24, Figures 25, and Figures 26, we can observe that the infrared images generated by the Semantic-Constrained StarGAN network for each time period extension are realistic, retaining the scene information of the original infrared images while also effectively extending the image to the corresponding infrared image domain for that time period. Compared to the experimental results obtained in Section 3.1.4 using the same scenes, Semantic-Constrained StarGAN shows a significant improvement in generation performance.

4 Concluasion

In this paper, we address the task of non-matching infrared image time period extension. We begin by analyzing the rationale for using the StarGAN network in this context, followed by a review of the experimental results. We then examine the challenges, including inconsistent extension outcomes for the same material and inaccuracies in time period extensions of images after StarGAN processing. To resolve these issues, we propose an infrared image time period extension method based on the Semantic-Constrained StarGAN network. This

approach incorporates semantic constraints during network training by integrating scene semantic segmentation information and a semantic encoding consistency loss. Additionally, we introduce a method for encoding scene time period information using semantic segmentation maps and infrared texture time-varying curves. Experimental results demonstrate that the Semantic-Constrained StarGAN network performs more effectively for infrared image time period extension tasks.

References

[1] P. A. M. Jacobs, "Simulation of the thermal behavior of an object and its nearby surroundings," *Journal Name*, 1980.

[2] N. Ben-Yosef, K. Wilner, S. Simhony, *et al.*, "Measurement and analysis of 2-d infrared natural background," *Applied Optics*, vol. 24, no. 14, pp. 2109–2113, 1985.

[3] H. Biesel and T. Rohlfing, "Real-time simulated forward looking infrared (flir) imagery for training," in *Infrared Image Processing and Enhancement*, vol. 781, pp. 71–81, International Society for Optics and Photonics, 1987.

[4] J. P. Gambotto, "Combining image analysis and thermal models for infrared scene simulations," in *Proceedings of 1st International Conference on Image Processing*, vol. 1, pp. 710–714, IEEE, 1994.

[5] M. J. Cathcart, "Generation and application of high-resolution infrared computer imagery," *Optical Engineering*, vol. 30, no. 11, p. 1745, 1991.

[6] H. K. Hong, S. H. Han, G. P. Hong, *et al.*, "Simulation of reticle seekers using the generated thermal images," in *Proceedings of APCCAS'96-Asia Pacific Conference on Circuits and Systems*, pp. 183–186, IEEE, 1996.

[7] T. Poglio, E. Savaria, and L. Wald, "Specifications and conceptual architecture of a thermal infrared simulator of landscapes," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 4540, pp. 488–497, 2001.

[8] A. W. Haynes, M. A. Gilmore, D. R. Filbee, *et al.*, "Accurate scene modeling using synthetic imagery," in *Targets and Backgrounds IX: Characterization and Representation*, vol. 5075, pp. 85–96, International Society for Optics and Photonics, 2003.

[9] J. Latger, T. Cathala, N. Douchin, *et al.*, "Simulation of active and passive infrared images using the se-workbench," in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XVIII*, vol. 6543, p. 654302, International Society for Optics and Photonics, 2007.

[10] T. Cathala, N. Douchin, A. Joly, *et al.*, "The use of se-workbench for aircraft infrared signature, taking into account body, engine, and plume contributions," in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXI*, vol. 7662, p. 76620U, International Society for Optics and Photonics, 2010.

[11] J. Mielikainen, B. Huang, and H. L. A. Huang, "Gpu-accelerated multi-profile radiative transfer model for the infrared atmospheric sounding interferometer," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 4, no. 3, pp. 691–700, 2011.

[12] J. Zhang, X. Fang, H. Zhang, W. Yang, and C. Zhu, "Statistical analysis of natural surface infrared radiation characteristics," *Infrared and Millimeter Waves Journal*, vol. 4, pp. 27–33, 1997.

[13] Z. Wang, Y. Lu, Q. Peng, Z. Jiang, and H. Zhu, "Infrared image formation model of urban buildings based on meteorology and heat transfer," *Journal of System Simulation*, vol. 5, pp. 517–523, 2000.

[14] Y. Xuan, D. Li, and Y. Han, "Infrared thermal image synthesis for complex ground backgrounds," *Infrared and Millimeter Waves Journal*, vol. 2, pp. 133–136, 2002.

[15] S. Chen, J. Sun, W. Guo, and L. Li, "A new method for generating infrared textures," *Journal of Dalian Maritime University*, vol. 36, no. 4, pp. 103–106, 2010.

[16] Q. Zhou, T. Bai, M. Liu, and C. Qiu, "Near-infrared scene simulation based on visible light images," *Infrared Technology*, vol. 37, no. 1, pp. 11–15, 2015.

[17] Y. Yang, M. Li, M. Yang, and Y. Wang, "Infrared texture generation method based on image interpolation modulation model," *Infrared Technology*, vol. 39, no. 3, pp. 214–220, 2017.

[18] Y. Wu and T. Zhang, "Computer simulation of infrared images and simulation software," *Infrared and Laser Engineering*, vol. 4, pp. 1–3, 2000.

[19] B. Da, N. Sang, and T. Zhang, "Long-wave infrared image simulation method using vega software," *Infrared and Laser Engineering*, vol. 3, pp. 333–337, 2007.

[20] M. Zhong, *Study on Automatic Generation Method of Infrared Scene Images Based on SE-Workbench*. PhD thesis, Huazhong University of Science and Technology, 2016.

[21] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2414–2423, 2016.

[22] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*, pp. 694–711, Springer, Cham, 2016.

[23] P. Isola, J. Y. Zhu, T. Zhou, *et al.*, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1125–1134, 2017.

[24] T. C. Wang, M. Y. Liu, J. Y. Zhu, *et al.*, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8798–8807, 2018.

[25] J. Y. Zhu, T. Park, P. Isola, *et al.*, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2223–2232, 2017.

[26] T. Kim, M. Cha, H. Kim, *et al.*, "Learning to discover cross-domain relations with generative adversarial networks," in *International Conference on Machine Learning*, pp. 1857–1865, 2017.

[27] Z. Yi, H. Zhang, P. Tan, *et al.*, "Dualgan: Unsupervised dual learning for image-to-image translation," in *International Conference on Computer Vision*, pp. 2849–2857, 2017.

[28] M. Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Advances in Neural Information Processing Systems*, pp. 700–708, 2017.

[29] X. Huang, M. Y. Liu, S. Belongie, *et al.*, "Multimodal unsupervised image-to-image translation," in *European Conference on Computer Vision*, pp. 172–189, 2018.

[30] Y. Choi, M. Ghoi, M. Kim, *et al.*, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *International Conference on Computer Vision and Recognition*, pp. 8789–8797, 2018.

[31] J. Xie, F. Li, H. Wei, and B. Li, "Infrared target simulation method based on generative adversarial networks," *Acta Optica Sinica*, vol. 39, no. 3, pp. 150–156, 2019.

[32] Y. Feng, *Research on Infrared Image Band Expansion Method Based on Actual Image*. PhD thesis, Xi'an University of Electronic Science and Technology, 2019.