RGB-TO-INFRARED IMAGE TRANSLATION FOR MILITARY AIRCRAFT USING CYCLEGAN - PROJECT PROGRESS REPORT

Bengisu YÜCEL
Department of Computer Engineering
Hacettepe University
Ankara, Turkiye
{bengisuyucel}@hacettepe.edu.tr

ABSTRACT

Infrared (IR) imaging provides critical capabilities for defense applications but suffers from limited dataset availability. This project implements CycleGAN for unpaired RGB-to-IR image translation, enabling synthetic IR generation without requiring aligned training data. I work on to adapt the original CycleGAN architecture with spectral normalization and identity loss to generate different domain figures for military aircraft dataset.

1 PROBLEM DEFINITION

Infrared (IR) imaging is critical for defense applications such as surveillance and reconnaissance, particularly in low-visibility environments. However, military-specific IR datasets remain scarce due to security constraints and collection challenges. This project addresses two interconnected problems:

- **Unpaired Translation**: Generating realistic IR images from RGB inputs without requiring pixel-aligned training pairs.
- **Domain Adaptation**: Adapting models trained on generic thermal datasets (e.g., urban scenes) to defense-specific domains (e.g., aircraft detection).

It is really hard to find IR images of military aircrafts to work on,CycleGAN solves the problem of unpaired image-to-image translation without needing extract pairs of corresponding images. We might have lots of images from both categories, but no direct correspondence between individual images.

2 RELATED WORK

In recent years, image-to-image translation using Generative Adversarial Networks (GANs) has become an active research focus, with significant advancements in model architectures and training strategies. A foundational work in this area is Pix2Pix by Isola et al. (2017), which introduced a supervised conditional GAN (cGAN) framework that learns to translate images from one domain to another using paired datasets. The Pix2Pix architecture consists of a single generator and discriminator pair trained with both adversarial and L1 losses, making it effective for aligned image pairs but limiting its applicability in domains where such pairs are difficult to acquire. To overcome this, CycleGAN by Zhu et al. (2017) proposed an unsupervised image-to-image translation framework that eliminates the need for paired data by introducing a dual generator-discriminator architecture and a novel cycle consistency loss. The model consists of two generators, one for each domain mapping, and two discriminators, with each generator aiming to fool its corresponding discriminator. The cycle consistency loss enforces that an image translated to the target domain and back should closely resemble the original, thereby preserving semantic content.

Building upon this dual structure, DualGAN by Yi et al. (2017) similarly employed a dual generator-discriminator setup but with a slightly different training strategy, leveraging a reconstruction loss alongside adversarial losses to stabilize unsupervised training. While both CycleGAN and DualGAN rely on cycle consistency, DualGAN emphasizes reconstruction quality within each domain, making it suitable for tasks like photo enhancement and artistic style transfer. StarGAN v2 by Choi et al. (2020) extended these ideas to a multi-domain translation framework using a single generator capable of handling multiple target domains with the help of a style encoder and mapping network. Unlike CycleGAN, which requires separate generators for each domain pair, StarGAN v2 simplifies architecture design for multi-class translation tasks.

In the thermal and infrared imaging domain, architectures have been specifically tailored for cross-modal translation. ThermalGAN by Wang et al. (2021) introduced a multimodal GAN model for translating color images to thermal images in person re-identification applications. This architecture incorporates identity preservation and multimodal discriminators to handle domain-specific challenges in thermal imaging. Similarly, CrossGAN by Liu et al. (2020) proposed a cross-domain translation network for agricultural imaging, leveraging adversarial and content losses to maintain cross-modal consistency. While these models share the adversarial learning principle with CycleGAN, they often integrate task-specific modules, such as attention mechanisms or feature alignment networks, to improve cross-modality translation.

For dataset resources, the FLIR ADAS Dataset (Systems, 2018) is widely used in autonomous driving systems for RGB-thermal paired image tasks, while the VEDAI dataset (Razakarivony & Jurie, 2016) offers aerial vehicle detection imagery valuable for small object detection, including thermal applications. These datasets provide essential benchmarks for evaluating models like CycleGAN and its variants. Collectively, these studies highlight both the architectural diversity and common foundational principles in GAN-based image translation, informing this project's approach to developing a CycleGAN-based RGB-to-infrared image translation model.

3 Method

In this project, we aim to develop an unpaired image-to-image translation model to convert visible spectrum (RGB) images of military aircraft into their corresponding infrared (IR) representations. The primary framework employed for this task is CycleGAN (Zhu et al., 2017), which enables unsupervised image translation by learning mappings between two domains without requiring paired images. Below, we outline the data preparation process and the model architecture used in detail.

3.1 CycleGAN Architecture

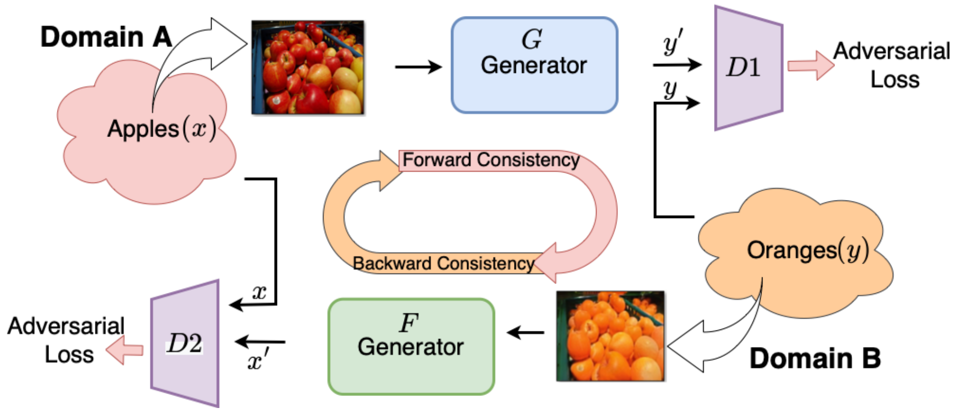

Figure 1: CycleGAN architecture overview. Figure adapted from (Rosebrock, 2022).

CycleGAN was proposed by Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros in their influential 2017 paper Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks (Zhu et al., 2017). This framework addressed one of the key limitations in early image-to-image translation systems: the requirement for paired training data. Most existing techniques at the time, such as Pix2Pix (Isola et al., 2017), relied on datasets where each input image had

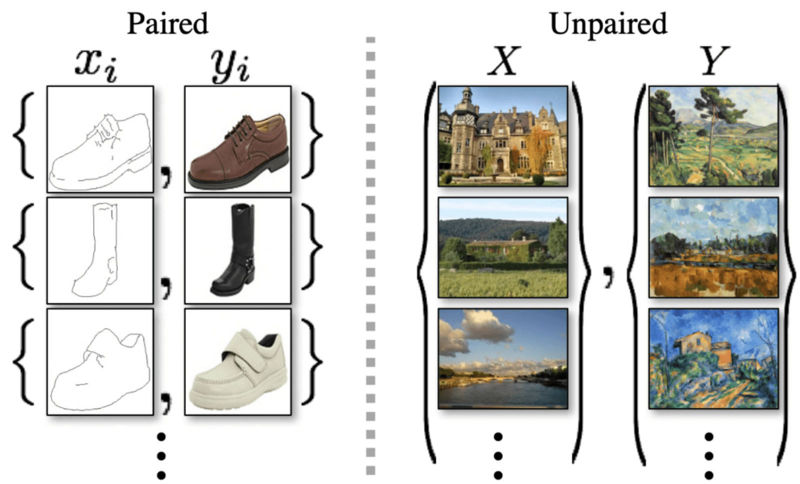

Figure 2: Sample depicting the case where data in the two domains is paired (left) and data is unpaired (right)(source: Zhu et al. (2017)

a corresponding target image, which is often difficult or expensive to obtain in many real-world domains. in input dimensions for model training.

4 Experimental Settings and Preliminary Results

4.1 Dataset and Setup

At first,Since I dont have any paired RGB-IR aircraft dataset, I'll consider to use paired RGB- IR dataset to train model. Then I examine VEDAI dataset.

- **VEDAI Dataset**: 1,920 unpaired RGB-IR training images, 572 test images.
- **Preprocessing**: Images resized to 256×256, normalized to $[-1, 1]$.
- **Augmentation**: Random crops, flips, and Gaussian noise.

Then I configured without paired data, I can still set up the network.Below, we outline the data preparation process and the model architecture used in detail.

Dataset Preparation To construct the dataset for this project, two separate unpaired collections of military aircraft images were prepared — one for RGB and one for IR images.

RGB Images: A total of 142 images were obtained from two sources. The primary source is the Military Aircraft Detection dataset available on Kaggle (Kaggle, 2024), which provides labeled images of various military aircraft classes. Additionally, several high-resolution RGB frames were manually cropped from military video footage available at military.com. The dataset covers over 70 distinct military aircraft types, including fighter jets, bombers, helicopters, and reconnaissance aircraft such as F-35, B-2, C-130, Mi-24, and AH-64.

Infrared (IR) Images: The IR dataset consists of 121 images sourced from the IR Dataset on Roboflow (Roboflow, 2024). These images represent various aerial vehicles captured in infrared spectrum imagery. The IR images are unpaired with the RGB images, making them suitable for unsupervised translation using CycleGAN.

Each image was resized and standardized to ensure consistency in input dimensions for model training. The core model architecture used in this project is CycleGAN. I adopt CycleGAN's standard framework (Figure 1) with:

- **Two Generators:**

  - Generator $G : X \rightarrow Y$ translates RGB images to IR images.
  - Generator $F : Y \rightarrow X$ performs the inverse operation, translating IR images back to RGB images.
- **Two Discriminators:**
  - Discriminator D_Y distinguishes real IR images from translated IR images generated by G.
  - Discriminator D_X distinguishes real RGB images from translated RGB images generated by F.

The model is trained using the following loss functions:

- **Adversarial Loss:** Ensures that each generator produces visually convincing images in the target domain by competing against the corresponding discriminator.
- **Cycle Consistency Loss:** Enforces that an image translated to the target domain and then back to its original domain returns a result close to the original image. Mathematically:

$$F(G(x)) \approx x \quad \text{and} \quad G(F(y)) \approx y$$

- **Identity Loss (optional):** Encourages the generator to preserve image characteristics when an image is passed through its own domain mapping, i.e.,

$$G(y) \approx y \quad \text{and} \quad F(x) \approx x$$

4.2 Implementation Challenges

Throughout the development of this project, several practical challenges were encountered during the implementation of the CycleGAN framework for RGB-to-IR image translation. One of the primary issues was **data scarcity**, as obtaining military-grade infrared image datasets suitable for fine-tuning and experimentation proved difficult. Publicly available datasets such as VEDAI were considered, but these lacked domain-specific features relevant to defense and military aerial imagery, creating a significant **domain mismatch** problem.

On the hardware side, the project faced notable constraints. Due to **GPU memory limitations**, the training process frequently crashed when using a batch size greater than 1 on an NVIDIA RTX 3060 GPU running under an Ubuntu WSL environment. Additionally, configuring the software environment introduced further complications, with multiple **dependency conflicts** arising between different versions of PyTorch, CUDA, and other supporting libraries, which occasionally led to kernel errors and runtime failures.

4.3 Training Implementation Code of Setup Network

Listing 1: InfraGAN Training Script with CUDA Fixes

```
import time
from options.train_options import TrainOptions
from data.data_loader import CreateDataLoader
from models.models import create_model
from util.visualizer import Visualizer

# CUDA Stability Fixes
torch.backends.cudnn.enabled = False  # Disable cuDNN
torch.backends.cudnn.benchmark = False
os.environ['CUDA_LAUNCH_BLOCKING'] = "1"  # Synchronous execution

# Initialize
opt = TrainOptions().parse()
opt.batchSize = 1  # Force batch size=1 for stability
opt.loadSize = 128  # Reduced from 256
opt.fineSize = 128
```

```
model = create_model(opt)
data_loader = CreateDataLoader(opt)
dataset = data_loader.load_data()

# Main Training Loop
for epoch in range(opt.epoch_count,
                   opt.niter + opt.niter_decay + 1):

    epoch_start = time.time()

    for i, data in enumerate(dataset):
        model.set_input(data)

        try:
            model.optimize_parameters()  # Core CUDA ops
        except RuntimeError as e:
            print(f"CUDA Error: {e}")
            torch.cuda.empty_cache()
            continue

        # Logging
        if i % opt.print_freq == 0:
            errors = model.get_current_errors()
            print(f'Epoch {epoch}, Iter {i}: {errors}')

    # Validation
    if epoch % 5 == 0:
        ssim_val = evaluator.eval(model)
        print(f'Validation SSIM: {ssim_val:.4f}')

    # Save checkpoint
    if epoch % opt.save_epoch_freq == 0:
        model.save(f'epoch_{epoch}')
```

Key Modifications

- **Line 6-8**: Added CUDA stability fixes
- **Line 11-13**: Enforced memory-safe parameters
- **Line 23-26**: Error handling for CUDA ops

All progress to date has been self-driven and implemented individually. To mitigate these issues, several workarounds were applied. The batch size was reduced to 1 to fit within the available GPU memory limits, and `torch.backends.cudnn.benchmark` was disabled to prevent further instability during training. Recognizing the limitations of the local setup, the project was subsequently migrated to **Google Colab Pro**, leveraging its access to high-memory GPUs and pre-configured environments to streamline the training process and avoid local hardware bottlenecks. This move significantly improved the training stability and allowed for extended experimentation with different hyperparameters and model configurations.

References

Yunjey Choi, Youngjung Uh, Jungkwon Yoo, and Jaejun Ha. Stargan v2: Diverse image synthesis for multiple domains. *CVPR*, 2020.

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 (CVPR)*, pp. 1125–1134, 2017. doi: 10.1109/CVPR.2017.632.

Kaggle. Military aircraft detection dataset, 2024. URL https://www.kaggle.com/datasets/a2015003713/militaryaircraftdetectiondataset.

Jie Liu, Xinchao Wang, Ming Li, Weidong Chen, and Jian Zhang. Crossgan: Cross-domain generative adversarial network for agricultural image translation. *IEEE Access*, 8:117527–117540, 2020. doi: 10.1109/ACCESS.2020.3004786. URL https://ieeexplore.ieee.org/document/9127641.

Salim Razakarivony and Frederic Jurie. Vehicle detection in aerial imagery: A small target detection benchmark. *Journal of Visual Communication and Image Representation*, 2016.

Roboflow. Ir dataset, 2024. URL https://universe.roboflow.com/ir-lq0tw/ir-dataset/browse?queryText=&pageSize=50&startingIndex=0&browseQuery=true.

Adrian Rosebrock. Cyclegan: Unpaired image-to-image translation, 2022. URL https://pyimagesearch.com/2022/09/12/cyclegan-unpaired-image-to-image-translation-part-1/. Accessed: 2025-05-03.

FLIR Systems. Flir adas dataset, 2018. URL https://www.flir.com/oem/adas/.

Guile Wang, Tianyi Zhang, and Jian Cheng. Thermalgan: Multimodal color-to-thermal image translation for person re-identification. *IEEE Sensors*, 2021.

Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2849–2857, 2017. doi: 10.1109/ICCV.2017.310.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *ICCV*, 2017.