Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Andreas Øie

Unsupervised RGB-to-Infrared Image Translation with Generative Models

Master's thesis in Cybernetics and Robotics
Supervisor: Håkon Hagen Helgesen
Co-supervisor: Øystein Kaarstad Helgesen
June 2023

NTNU
Norwegian University of
Science and Technology

Andreas Øie

Unsupervised RGB-to-Infrared Image Translation with Generative Models

Master's thesis in Cybernetics and Robotics
Supervisor: Håkon Hagen Helgesen
Co-supervisor: Øystein Kaarstad Helgesen
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Abstract

This thesis investigates the application of unpaired image-to-image translation techniques and generative models for generating synthetic data to enhance the milliAmpere 2's autonomous systems. The study focuses explicitly on generating infrared (IR) images from electro-optical (RGB) images, enabling improved maritime object detection and recognition. A comprehensive literature review and experimental evaluation of existing state-of-the-art unpaired image translation methods and relevant strategies were performed, leading to the selection of CycleGAN as the most suitable model for this task.

An extensive amount of time was devoted to testing and implementing different types of generative models, as well as exploring configurations to achieve optimal performance. The dataset creation process involved addressing the challenges arising from the data exploration and previous discoveries from the Specialization Report. The baseline models, CycleGAN, CUT, GcGAN, and StarGAN-v2, were implemented and evaluated using the quantitative Fréchet Inception Distance (FID) metric and a qualitative custom visual evaluation scheme. CycleGAN emerged as the top-performing model, generating the best images based on visual quality for this dataset.

The results from the various tuning experiments showed a modified CycleGAN model with a ResNet-based architecture in the generator, Xavier weight initialization, ReLU as the non-linear activation, and a learning rate of $1e^{-4}$ to be the best configuration. This modified model generated the most realistic infrared images, reaching a visual evaluation score of 3.75/5 and an FID score of 135. These results indicate a direct improvement compared to the Specialization Report, where an FID score of 195 was reached. However, the model's performance was still limited by object imbalances and inherent dataset challenges, such as object imbalance, available images, resolution, and the field-of-view difference between the electro-optical and infrared camera outputs.

In conclusion, the findings demonstrate the feasibility of leveraging synthetic data for improving autonomy system performance in maritime applications while shedding light on potential research directions. Future work should focus on expanding the dataset, addressing the challenges within the dataset, exploring recent trends within the space of generative models, and investigating evaluation metrics suitable for infrared imagery. The research conducted in this thesis provides a foundation for further improvements and refinements to the unpaired image-to-image translation methods, which could become a valuable tool for incorporating infrared cameras into the milliAmpere 2's autonomy system, improving the vessel's situational awareness during nighttime and in poor weather conditions.

Sammendrag

Denne masteroppgaven undersøker anvendelsen av uparede bilde-til-bilde oversettelsesteknikker og generative modeller for å generere syntetiske data for å forbedre milliAmpere 2's autonome systemer. Arbeidet fokuserer eksplisitt på å generere infrarøde (IR) bilder fra elektro-optiske (RGB) bilder, noe som muliggjør forbedret deteksjon og gjenkjenning av maritime objekter. En omfattende litteraturgjennomgang og eksperimentell evaluering av eksisterende toppmoderne uparede bildeoversettelsesmetoder og relevante strategier ble utført, noe som førte til valget av CycleGAN som den mest egnede modellen.

En betydelig mengde tid ble viet til testing og implementering av flere ulike modeller og utforsking av konfigurasjoner for å oppnå optimal ytelse. Opprettelsen av et nytt datasett involverte å håndtere utfordringene som oppstod fra datautforskningen og tidligere funn fra forprosjektet. Referansemodellene, CycleGAN, CUT, GcGAN og StarGAN-v2, ble implementert og evaluert ved hjelp av den kvantitative Fréchet Inception Distance (FID) metrikken og et kvalitativt tilpasset visuelt evalueringsskjema. CycleGAN fremsto som den beste modellen, og genererte de beste bildene basert på realisme og bildekvalitet for dette datasettet.

Resultatene fra de ulike tuningseksperimentene indikerte at en modifisert CycleGAN-modell med en ResNet-basert arkitektur i generatoren, Xavier vektinitialisering, ReLU som ulineær aktiveringen, og en læringsrate på $1e^{-4}$ som den beste konfigurasjonen. Denne modifiserte modellen genererte de mest realistiske infrarøde bildene, og oppnådde en visuell evalueringsscore på 3,75/5 og en FID-score på 135. Disse resultatene viser til en direkte forbedring sammenlignet med forprosjektet, der en FID-score på 195 ble oppnådd. Imidlertid var modellens ytelse fortsatt begrenset av utfordringer med datasettet, som skjevfordeling av objekttyper, antall tilgjengelige bilder, bildeoppløsning og ulikheter i synsfelt mellom de elektro-optiske og infrarøde kameraene.

Avslutningsvis viser funnene gjennomførbarheten av å utnytte syntetiske data for å forbedre autonomisystemets ytelse i maritime applikasjoner, samtidig som det kaster lys over potensielle forskningsretninger. Fremtidig arbeid bør fokusere på å utvide mengden på datasettet, håndtere utfordringene i datasettet, utforske nyere trender innen generative modeller og undersøke evalueringsmetrikker som passer bedre for bilder i gråskala. Denne oppgaven har lagt grunnlaget for videre forbedringer av uparede bilde-til-bilde oversettelsesmetoder, noe som kan bidra til å optimalisere bruken av infrarøde kameraer i milliAmpere 2's autonomisystem og dermed forbedre fartøyets evne til å navigere og oppfatte omgivelsene under nattlige og utfordrende værforhold.

Preface

This thesis represents the culmination of my academic journey in the Cybernetics and Robotics Master's Degree Programme at the Norwegian University of Science and Technology (NTNU). The work presented in this thesis, completed during the Spring of 2023 as part of the TTK4900 - Engineering Cybernetics, Master's Thesis course, is a natural continuation and extension of the research conducted in my previous report for the subject TTK4551 - Engineering Cybernetics, Specialization Project [38].

The subject of this thesis relates to developing and implementing deep-learning algorithms for autonomous ferry transportation, with a particular focus on synthesizing infrared images from visible-spectrum images. It builds upon the collaboration between NTNU and Zeabuz, a spin-off from the NTNU research community focused on autonomous vessels.

The reader is assumed to have a background in computer science, engineering, or a related field and a fundamental understanding of deep learning and computer vision concepts. Familiarity with generative adversarial networks (GANs) and image-to-image translation techniques would be beneficial for fully appreciating the content of this thesis.

I want to thank my supervisors, Håkon Hagen Helgesen and Øystein Kaarstad Helgesen, for their continued support and encouragement throughout this thesis.

Contents

1 Introduction

1.1 Background and Motivation

In an increasingly digitalized world, the development of autonomous electric vehicles and vessels, a constantly evolving field, contributes to reducing the overall impact of transportation on the environment while enhancing efficiency and safety. Numerous sectors within the transport and mobility domain can mitigate the environmental footprint caused by transportation on a global scale. The history of transportation has evolved from fossil-fuel-powered vehicles to electric vehicles (EVs) - developed to reduce pollution and emissions.

Advancements in Airborne, Land-based, and Maritime transportation

Commercial airplanes have adopted advanced autopilot systems in airborne transportation, controlling various aspects such as altitude, speed, and direction during most flight phases. Pilots typically handle takeoff and landing manually, although advanced systems can assist. Despite the efforts to automate airborne transportation, human pilots remain crucial for safety and efficiency.

In the land-based transportation sector, many standard vehicles, such as buses and cars, have been leading the path in terms of transportation. Regarding travelling between destinations, private cars still reign on top as the most commonly used transport method. Industry car brands continue to adopt the latest and most remarkable technologies to improve their capability. Tesla, a leading actor, has developed their autonomous driver-assistance system (ADAS) - called Autopilot[1]. ADAS systems are a recent breakthrough in the transport domain and are a profound step towards expanding the boundaries of autonomous transportation.

In the maritime environment, similar analogies can be made for autonomous driving. Larger vessels such as container ships and cruise ships are using self-driving assistance systems, such as Yara Birkeland[2]. As technology advances, the potential for entirely end-to-end autonomous maritime operations increases, bringing benefits such as improved safety, efficiency, and reduced environmental impact. Various companies are destined to tackle this challenge of developing autonomous control and advanced perception systems for marine vessels. Honourable mentions such as the US company Sea Machines Robotics[3], the Finnish company Wärtsilä[4], the south Korean Avikus[5], and the Norwe-

[1] Visit https://www.tesla.com/autopilot for more information

[2] See info at https://www.yara.com/news-and-media/media-library/press-kits/yara-birkeland-press-kit/

[3] See details at https://sea-machines.com/ai-powered-vessel-vision/

[4] See more at https://www.wartsila.com/voyage/autonomy-solutions

[5] Visit https://avikus.ai/eng/product/hinas for more information

gian Kongsberg Maritime[6] are all actively researching this area. Another honourable mention from Norway is Zeabuz (in collaboration with NTNU). In late 2022, Zeabuz and NTNU demonstrated the world's first urban autonomous passenger ferry, milliAmpere 2, capable of transporting up to 20 people simultaneously.

The Role of Cameras

The milliAmpere 2 ferry, owned by NTNU, visualized in Figure 1.1, is a battery-powered and environmentally friendly transportation vehicle. The vessel has various sensors and computing hardware, including RTK GNSS (Real-Time Kinematic Global Navigation Satellite System), radar, lidar, electro-optical, and infrared cameras. The autonomy system, developed by Zeabuz, comprises four core parts[7]: See, Understand, Plan, and Act. In this thesis, the research is motivated by cameras' vital role in the See and Understand components. While radar and lidar work in tandem to detect and track objects in the maritime environment, cameras can capture helpful information such as color, texture, shape, size, type of object, and movement, offering a more comprehensive and detailed visual understanding.

Infrared Cameras and Their Potential

In the maritime environment, infrared cameras provide valuable information during nighttime and in poor weather conditions. Combined with electro-optical cameras, they provide essential input for the planning and execution of safe and efficient routes while navigating between destinations. By continuously reevaluating its planned path and speed and considering the images captured by these cameras, the milliAmpere 2 ferry can apply appropriate adjustments based on situational awareness. Unlike their counterparts that capture visible light, thermal cameras offer visibility in the infrared spectrum. They effectively complement visible spectrum cameras, proving particularly useful for tasks such as robot localization and navigation in visually challenging environments [24]. For example, as one can observe from Figure 1.2, the information captured with one electro-optical and one infrared camera is quite significant.

Addressing Data Imbalance with Synthetic Data

While the current autonomy system is a significant step towards the complete autonomy of the milliAmpere 2, there is still room for improvement. The current detection system deployed on the ferry needs to be more robust to handle all possible scenarios across all weather conditions. Infrared cameras present a potential visual enhancement, as they are less sensitive to lighting conditions and can detect objects with large thermal gradients, such as swimmers or kayakers. This makes infrared cameras highly suited for certain

[6] See more at https://www.kongsberg.com/maritime/products/bridge-systems-and-control-centres/navigation-systems/autopilot/

[7] Find out more at https://www.zeabuz.com

Figure 1.1: Overview of milliAmpere and its camera components. Left-image shows the ferry itself. Top-right shows the electronic box capsuling the various EO and IR cameras. Bottom-right shows the dual-camera setup of a single angle with the infrared lens at the top and electro-optical at the bottom. Left-image is sourced with approval from Universitetsavisa by photographer Oda Eriksen Haugland.

types of object detection in the maritime environment and could significantly enhance the safety and robustness of the milliAmpere 2's autonomy system. The current autonomy does not use infrared cameras as of 12th May 2023; hence enabling such cameras would, in addition to mentioned advantages, act as a redundancy guarantee, in conjunction with the current electro-optical camera system.

The Growing Role of Synthetic Data in Various Fields

This thesis aims to explore the use of synthetic data to improve the performance of the milliAmpere 2's autonomy system. The vast majority of the open-world image datasets have been captured during daytime under ideal conditions in the visual spectrum. There is limited maritime traffic during nighttime or in poor weather conditions. Consequently, there is a significant lack of infrared images and datasets that can be utilized for training the detection system to handle such situations. The thesis seeks to address this data imbalance by employing existing daylight images and translating them to the infrared spectrum using generative models. Although there have been notable advancements

Figure 1.2: Illustration showcasing the visibility of the boat taken with an electro-optical (left-side) and infrared (right-side) camera in pitch-dark environments. Image is sourced from [16] with permission.

in unpaired image-to-image translation, including near-infrared to RGB and thermal infrared to visible color tasks, there remains a gap in the literature concerning direct RGB to IR translation, the exact focus of this thesis. Using deep learning techniques to automatically generate infrared images from RGB images, one could leverage considerably larger data resources which, in turn, can be used to retrain detection systems.

The Importance of AI and Synthetic Data

Synthetic data has been used for a long time in various fields, including computer graphics, animation, and gaming. In recent years, the adoption of synthetic data in fields such as robotics, self-driving cars, and medical imaging has led to significant success. Synthetic data can be generated using methods like procedural generation, physics-based simulations, and deep learning techniques like generative adversarial networks (GANs). An NVIDIA blogpost[8] cites a Gartner study on synthetic data, which predicts that by 2030, the majority of data utilized in AI will be artificially created using methods such as rules, statistical models, simulations, and other techniques. The potential solution to the problem of limited infrared images in the maritime environment could be addressed by generating synthetic infrared images, which could provide a larger dataset for training object detection models. Synthetic data has several advantages, including creating diverse and balanced datasets and eliminating privacy concerns. However, it may not always accurately represent real-world scenarios, and its usefulness depends on the quality of the generated data. The importance of AI in the modern world cannot be overstated, as it has become integral to a wide range of industries and applications. The synergy between various techniques, such as machine learning, computer vision, and generative models, fosters innovation and drives the progress of advanced systems across multiple domains.

[8] See https://blogs.nvidia.com/blog/2021/06/08/what-is-synthetic-data/

Unpaired Image-to-Image Translation

Adopting synthetic data in autonomous systems is a promising development, as these systems rely heavily on robust detection systems for safe and efficient operation. Research in computer vision and deep learning has enabled the generation of synthetic data, potentially improving various systems' performance. Synthetic data is created primarily using generative models and image-to-image translation techniques, which offer unique capabilities for diverse applications.

At the core of this research topic is unpaired image-to-image translation, a concept closely related to the context described above. It involves synthetically converting data from a source domain to a target domain without paired examples. Achieving accurate translation could significantly impact the industries mentioned, other researchers, and the overall effectiveness of autonomous systems. In this thesis, enabling a model to generate realistic synthetic infrared images from RGB images could provide a valuable resource for enhancing the detection capabilities of future autonomous systems.

1.2 Research Objectives

- **Literature Review**: Conduct a comprehensive survey of existing unpaired image translation methods and relevant strategies. Based on this review, evaluate whether the techniques studied in the Specialization Project [38] should be further explored or if alternative methods hold greater promise. Make informed decisions regarding selecting methods for further investigation and experimental testing.
- **Method Implementation**: Identify the most suitable method(s) for the task and determine an appropriate testing environment, considering both remote and local resources. Implement the selected method(s) for experimental evaluation, emphasizing the need for longer training durations and choice of hyperparameters.
- **Dataset Assessment and Expansion**: Evaluate the existing dataset, considering factors such as the presence of foreground objects and the environments in which they were captured. Based on this assessment, create a new and more specific dataset.
- **Case Study Design and Execution**: Plan and conduct relevant case studies employing the chosen methods and strategies. Utilize applicable data from the maritime domain, focusing on the environment in which smaller passenger ferries operate.
- **Qualitative Performance Analysis**: Perform a qualitative and quantitative analysis of the experimental performance using data from milliAmpere's sensor rig, which incorporates co-localized infrared and daylight cameras. Evaluate the effectiveness of the selected method(s) on industry-standard evaluation metrics, and identify potential areas for improvement.

1.3 Contributions

- A thorough exploratory analysis of all available data from the milliAmpere's sensor rig.
- The creation of a refined and condensed dataset consisting of only images with foreground objects in focus during maritime traffic.
- A proper investigation using various state-of-the-art generative adversarial networks.
- An extensive experimental comparison of four generative adversarial networks, where each model has been configured, trained and evaluated on industry-standard evaluation metrics.
- A broad exploration of various hyperparameters used to improve the realism and quality of the generated images.
- A customizable framework for future research and applications in the field of computer vision and generative models, allowing for easy integration and adaptation of new techniques and models.
- An intuitive web application for manual image classification.

1.4 Report Outline

- **Chapter 2: Background Theory and Related Work** - This chapter provides an overview of the fundamental concepts and previous research in the field of generative models.
- **Chapter 3: Methods** - This chapter describes the research design, data sources, and analytical techniques employed to address the research objectives.
- **Chapter 4: Implementation** - This chapter outlines the practical realization of the proposed research, explaining the process of model development, optimization, and training.
- **Chapter 5: Results and Discussion** - This chapter presents the research findings, assesses the developed models' performance and discusses the implications of the results in both theoretical and practical aspects.
- **Chapter 6: Conclusion** - This chapter summarizes the key findings, discusses the study's limitations, and provides recommendations for future research.

2 Background Theory and Related Work

This chapter provides the reader with an overview of the essential background theory and related work that establish the research presented in this thesis. The chapter begins by discussing the fundamental concepts of machine learning and deep learning and highlighting the key differences between them, followed by an introduction to generative models, focusing on deep generative models and their corresponding architectures, which are crucial for understanding the nuances of image translation.

For transparency, some theoretical details about GANs and Image-to-Image translation, in Sections 2.2.2 and 2.3, are taken from the Specialization Report [38].

2.1 Deep Learning

Deep learning is a sub-field of machine learning that focuses on algorithms inspired by the structure and function of the human brain, called artificial neural networks. While machine learning consists of various techniques and algorithms, deep learning involves using deep neural networks with multiple hidden layers. Due to its ability to model complex data and abstract features, deep learning has advanced several domains, including computer vision, natural language processing, and reinforcement learning.

The key difference between classical machine learning and deep learning is their architecture and applicability. Classical machine learning techniques, such as linear regression, support vector machines, and decision trees, do not rely on layers like deep learning algorithms do. Instead, they have different structures and methodologies for learning patterns from primarily structured data. They can be powerful and effective for specific tasks but may be less suitable when dealing with unstructured data such as images or sounds.

Deep learning is characterized by deep architectures, such as deep neural networks, consisting of multiple layers (input layer, multiple hidden layers, and output layer). This enables deep learning models to capture intricate relationships and learn abstract representations from large datasets, making them suitable for tasks that require high-level feature extraction and complex decision-making. Deep learning has been the driving force behind significant advances in AI, such as the development of OpenAI's generative languages models ChatGPT [40] and its successor GPT-4 [39], or the state-of-the-art text-to-image models Stable Diffusion [46] from Stability AI and Midjourney from Midjourney's research lab [34]. Deep learning models can be costly in computation, necessitate substantial training data, and are occasionally perceived as black-box models because of their complex inner workings.

2.1.1 Fundamentals

In generalized terms, deep neural networks comprise multiple layers of interconnected artificial neurons or nodes. Each layer receives input from the previous layer, performs

a non-linear transformation, and passes the output to the subsequent layer. A deep neural network typically has many hidden layers, allowing it to learn and represent more intricate and sophisticated features from the data, thus enhancing its problem-solving capabilities.

Training deep neural networks involves a process called backpropagation, which adjusts the weights of the connections based on the errors between the predicted and actual output. This optimization uses gradient descent and its variants, which minimize a specified loss function. For instance, adjusting the learning rate, a hyperparameter controlling the update step size is commonly fine-tuned to achieve an optimal trade-off between convergence speed and stability.

Deep learning has several additional hyperparameters that influence the model's performance, such as the number of neurons per layer, activation functions, and regularization techniques. The choice of these hyperparameters can significantly impact the model's ability to generalize to unseen data and prevent overfitting.

2.1.2 Architectures

Multi-Layer Perceptrons (MLPs) are fundamental deep learning architectures with multiple layers of fully connected neurons. They connect each neuron in one layer to every neuron in the subsequent layer, with non-linear activation functions enabling the modeling of complex relationships. MLPs are versatile and can be used for various tasks, such as classification and regression, making them a foundation for understanding more advanced deep learning architectures.

Convolutional Neural Networks (CNNs) are favored for computer vision tasks. They are unique due to their use of convolutional layers, which learn local spatial patterns in input data, such as images. These layers apply filters to the input, allowing CNNs to effectively capture hierarchical features in visual data. As a result, CNNs excel in tasks like image classification, object detection, and semantic segmentation.

Recurrent Neural Networks (RNNs) are designed for sequential data and natural language processing tasks. What sets them apart is their internal memory cells, which maintain information from previous time steps, letting them model temporal dependencies in data. RNNs are particularly suitable for tasks involving time-series data, text generation, and sentiment analysis, as they can capture patterns and relationships across sequences.

Recently, transformer-based architectures have gained popularity due to their ability to model long-range dependencies in sequential data, outperforming RNNs and CNNs in specific language processing tasks. Transformers employ self-attention mechanisms, focusing on different parts of the input sequence, which has led to their success in machine translation, text summarizing, and other natural language understanding tasks.

Specialized architectures such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) have been developed for unsupervised learning tasks, including image synthesis and data generation. These generative models learn to capture

the underlying data distribution and generate new samples resembling the original data, enabling applications from image-to-image translation to anomaly detection.

2.2 Generative Models

Historically, generative models relied on statistical methods such as Kernel Density Estimation[1] and Gaussian mixture models[2]. However, they often produced suboptimal results due to the assumptions[3] made about the underlying distributions in the datasets.

With the emergence of deep learning, deep generative models have gained considerable attention and become highly influential in generative modeling. These models, driven by deep neural networks, demonstrate remarkable performance in learning complex data distributions. Autoencoders (AE), first introduced in the 1980s [48], and Variational Autoencoders (VAE), introduced in 2013 [25], are widely used deep generative models that learn a compact, lower-dimensional representation of the data and generate new samples by sampling in the learned latent space. These models can generate high-quality images, text, and other data types. Generative models aim to learn the underlying distribution of a given dataset and generate new samples that share similar characteristics. Discriminative models, conversely, learn a decision boundary between classes and classify new data points based on that boundary.

2.2.1 Architectures

Recent trends in deep generative models have led to the development of new architectures such as Generative Adversarial Networks (GANs), famously introduced by Ian J. Goodfellow [13], Flow-based Generative Models, and Diffusion Models. These deep generative models have demonstrated impressive capabilities in synthesizing realistic and diverse samples, further emphasizing the role of deep neural networks in advancing the field of generative modeling.

[1] A non-parametric method for estimating the probability density function of a random variable using a weighted sum of kernel functions

[2] A probabilistic model representing a mixture of multiple Gaussian distributions to describe the underlying distribution of a dataset

[3] Assumptions regarding specific parametric distributions, feature independence, linearity, and homoscedasticity

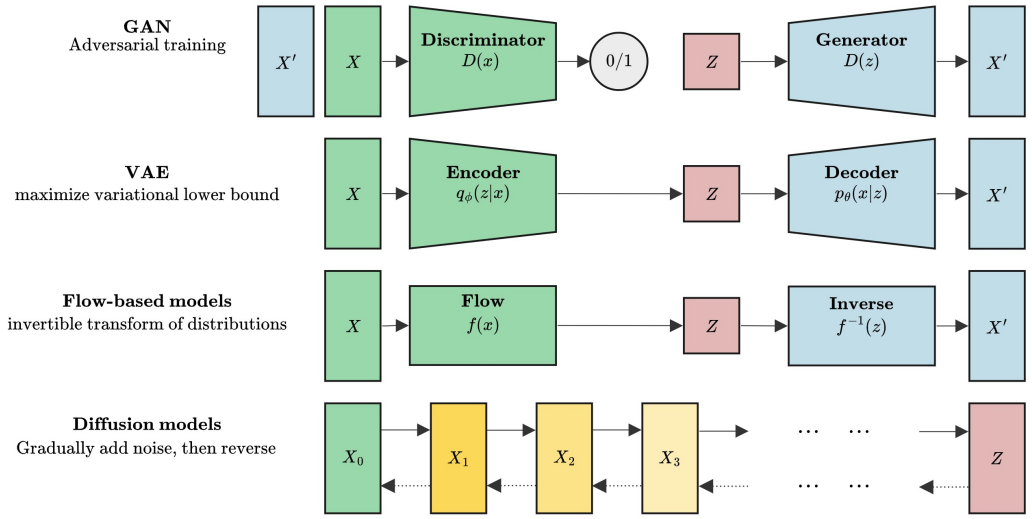


Figure 2.1: Overview of different types of Generative Models.

GANs, VAEs, Flow-based, and Diffusion-based models, as visualized in Figure 2.1, are all generative models but differ in the methods and architecture they use to synthesize data.

GANs, initially introduced by Ian J. Goodfellow [13], are deep neural networks that consist of two parts, a generator network and a discriminator network, that are trained together in a competitive game. The generator network produces new data samples, while the discriminator network tries to distinguish between real data and the samples generated by the generator. GANs have been used to generate realistic images, videos, and even audio.

VAEs [26][45] are an extension of autoencoders, where they optimize a lower bound on the data likelihood using variational inference. Unlike autoencoders, VAEs learn a probabilistic mapping between the data and latent spaces by introducing a Gaussian prior to the latent variables. This prior helps VAEs generate more realistic and diverse samples than regular autoencoders, thus making VAEs particularly suitable for generative modeling tasks.

Flow-based models [10][11] are another class of generative models that learn an invertible mapping from the data to a latent space, which can be sampled to produce new data. This mapping ensures that the generated data samples are well-represented by the learned distribution. Flow-based models have demonstrated state-of-the-art results in generating high-quality images.

Diffusion models are a class of generative models that can model high-dimensional

distributions using the process of diffusion. Various concepts of diffusion processes for generative modeling have been proposed, such as diffusion probabilistic models [49], noise-conditioned score network [50], and denoising diffusion probabilistic models [17]. Diffusion models use an iterative sampling process from a diffusion process to generate new data samples. This process can generate high-fidelity images and videos that surpass the quality of other generative models.

While both supervised and unsupervised generative models exist, utilizing unsupervised methods like unsupervised image translation can help naturally deal with the absence of paired data, making them suitable for tasks like RGB-to-Infrared image translation, where obtaining perfect paired images could be challenging.

2.2.2 Generative Adversarial Networks

GANs aims to generate realistic data through an adversarial strategy involving two neural networks, as mentioned in Section 2.2. In the training process, the generator and discriminator engage in ongoing competition. The generator strives to produce more convincing fake samples, while the discriminator constantly refines its capacity to differentiate between actual and generated samples. This adversarial dynamic enhances the performance of both networks, ultimately leading to the generation of realistic data.

When using GANs for image tasks, the generator receives a random noise vector sampled from the latent space as input. The generator, consisting of for instance convolutional layers, normalization techniques, and activation functions, then uses the latent space noise to create something akin to an image. With a similar architecture, the discriminator processes both the generated image and a sample from the real image dataset, attempting to classify the authentic images by comparing them to the ground truth using a cost function. The parameters for both networks (generator and discriminator) are subsequently updated using backpropagation.

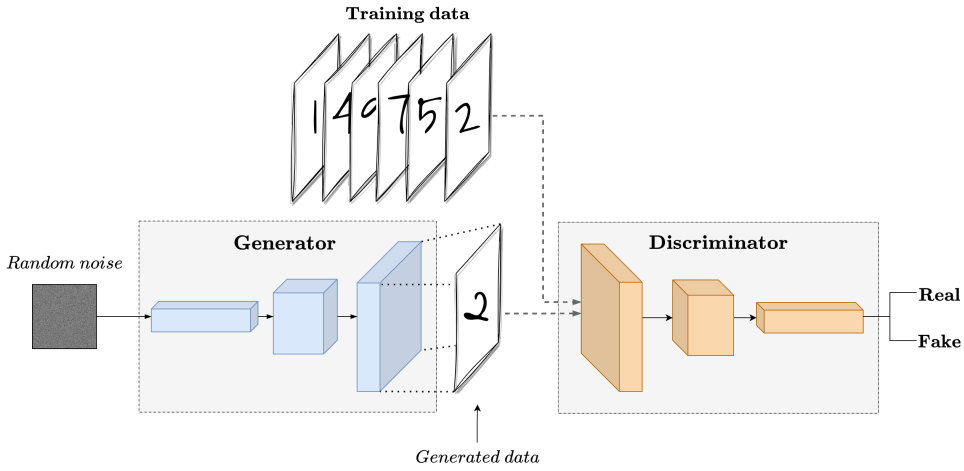


Figure 2.2: The illustrations portrays the learning process of a general Generative Adversarial Network (GAN), which involve two components - a generator and a discriminator. The generator attempts to generates samples similar to the real dataset. The discriminator is then asked to guess which of the two sets of images are real or fake.

Loss Functions

Loss functions play a significant role in the GANs learning process. The adversarial loss, a classical GAN function described below, theoretically makes generated images indistinguishable from actual images. This aligns with the primary goal of computer graphics, making the adversarial loss function particularly effective for tasks involving image creation. However, GAN training can be challenging due to mode collapse, vanishing gradients, or unstable convergence. Some techniques exist to help mitigate these issues, such as Wasserstein GANs [2] or Spectral Normalization [36].

Adversarial Loss

The adversarial loss function is used in the context of generative adversarial networks (GANs) to measure the discrepancy between the distributions of real and generated data. The GAN model has two parts, one for training the discriminator network and the other for training the generator network, each with its loss function. The interaction of these loss functions produces the adversarial loss function. The generator wants to minimize the objective, while the discriminator wants to do the opposite, maximize it, as shown by equation (2.1).

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))] \tag{2.1}$$

In this function:

- z, a random noise sample (latent vector), typically from a Gaussian or uniform distribution.
- $D(x)$, the discriminator's estimation of the probability that the real input data x is genuine.
- $G(z)$, the output of the generator when given noise z.
- $D(G(z))$, the discriminator's estimation of the probability that the generated input data $G(z)$ is genuine.
- $\mathbb{E}\boldsymbol{x} \sim p\text{data}\,(\boldsymbol{x})$, the expected value calculated over all instances of real input data.
- $\mathbb{E}\boldsymbol{z} \sim p\boldsymbol{z}(\boldsymbol{z})$, the expected value calculated over all instances of random input to the generator.

The generator's primary goal is to create images with a data distribution closely resembling the real data. Ideally, the real and generated data would be indistinguishable, causing the discriminator to make a random guess with a 50%-50% chance of being correct. This scenario occurs when the data distribution of both image types is so similar that they appear to have come from the same dataset. However, achieving this perfect alignment is rarely possible in practice, as training GANs are known to be a challenging task. Initially, the data distributions might differ significantly, but as the training progresses, the generated data distribution aligns more closely with the real data distribution.

Applications

GANs have demonstrated impressive results in various applications, such as transforming a scribble into a photo-like image [18] and converting footage of a horse into a running zebra [59], all without the need for laboriously annotated training data. This has led to significant advancements in various applications such as image synthesis, super-resolution, and data augmentation, to mention a few. In image synthesis, GAN architectures like Progressive GANs [23] and StyleGAN [22] generate high-quality, realistic images, such as artwork, faces, and scenery. Super-resolution GANs, like SRGAN [30], enhance the resolution of low-quality images, finding applications in video streaming, satellite imagery, and medical imaging. GANs can also be used as a data-augmentation technique to generate extra training data to improve machine learning model performance, particularly in situations with imbalanced datasets or when obtaining additional labelled data is challenging. These applications showcase the versatility of GANs across diverse research fields.

Key Architectures and their Advancements

In the vanilla implementation of the GAN architecture in 2014, the authors used Multilayer Perceptrons (MLP) for both the generator and the discriminator. MLPs are feed-

forward networks and are "linear" in their basic form, as they consist of multiple layers of nodes (neurons), where each layer is fully connected to the next one. Shortly after, architectures like ResNet [14], and U-Net [47] were incorporated in GAN design for more complex tasks. ResNet, short for Residual Networks, is characterized by its unique skip connections or shortcut connections, which allow the gradient to be directly backpropagated to earlier layers. On the other hand, U-Nets are a unique convolutional neural network architecture designed for efficient and accurate image segmentation. They employ a symmetric encoder-decoder structure, which enables precise localization of features, allowing the combination of high-resolution features from the contracting path with the expanding path. Over the years, several innovative architectures have been developed, each building on the original concept and introducing new techniques to improve performance, image quality, and domain-specific applications. Some of the key architectures include:

- Conditional GANs (2014): This architecture extends traditional GANs by conditioning the generator and discriminator on additional information such as class labels or text descriptions, enabling the model to generate images based on specific attributes [35].

- Deep Convolutional GANs (2015): This architecture replaces fully connected layers with convolutional neural networks (CNNs) in both the generator and discriminator, leading to more stable training and higher-quality images while maintaining the structure of the generated images [44].

- InfoGAN (2016): This architecture introduces an information-theoretic framework to GANs that maximizes the mutual information between a subset of the generator's input and the output, enabling the model to learn interpretable and disentangled representations [6].

- CycleGAN (2017): This architecture can learn to translate between two domains without paired examples, using cycle consistency loss to ensure that the mapping from one domain to the other and back is consistent, resulting in improved domain translation quality [59].

- Progressive GANs (2017): This architecture introduces a training procedure that generates high-resolution images by gradually adding layers to both the generator and discriminator during training, allowing the model to learn coarse-to-fine features and improving stability [23].

- StarGANs (2018, 2020): These architectures enable image-to-image translation across multiple domains using a single model, focusing on cross-domain translation for images. StarGAN-v2 addresses quality, diversity, and scalability issues present in the original StarGAN [7][8].

- StyleGANs (2018, 2020, 2021): These architectures introduce adaptive instance normalization (AdaIN) to control style at different levels of granularity, allowing for unprecedented control over the generated image's appearance and the ability to separate content from style [20][22][21].
- BigGAN (2019): This architecture scales up GANs using large-scale distributed training, achieving high-quality image generation at high resolutions, setting new benchmarks for image synthesis [5].
- Contrastive GANs (2020, 2021): These architectures incorporate contrastive learning techniques into the GAN framework, attempting to learn a contrastive loss function that encourages similar samples to be close together in latent space while pushing different samples apart, leading to improved image quality and diversity [42][52][19].

2.2.3 Evaluation Metrics

Evaluating the performance of generative models is challenging due to the nature of the generative models, which produce novel samples instead of replicating training data. Using effective and fair evaluation metrics is essential for evaluating GAN models, monitoring their training progress, and validating their usability in real-world scenarios. Some key evaluation metrics include Inception Score (IS), Structural Similarity Index (SSI), Learned Perceptual Image Patch Similarity (LPIPS), Fréchet Inception Distance (FID), and visual examination. In the landscape of metrics, one can observe their main properties in Tables 2.1, 2.2, 2.3, 2.4, and 2.5.

The following metrics are described and inspired by Ali Borji's two research experiments on "Pros and Cons of GAN Evaluation Measures" [3][4]

Inception Score (IS)	
Description	Measures the quality and diversity of generated images by utilizing a pretrained InceptionV3 model. It calculates the KL divergence between the conditional and marginal class distribution of generated samples.
Pros	Is simple to compute and can provide a quantitative measure of image quality and diversity
Cons	Does not capture intra-class diversity (variations of the same class in different images).

Table 2.1: Inception Score (IS): A quantitative metric for evaluating GANs

Structural Similarity Index Measure (SSIM)	
Description	Quantifies the structural similarity between two images by considering luminance, contrast, and structure.
Pros	More reliable than pixel-wise comparisons like MSE.
Cons	Sensitive to small structural misalignments.

Table 2.2: Structural Similarity Index Measure (SSIM): A quantitative metric for evaluating GANs

Learned Perceptual Image Patch Similarity (LPIPS)	
Description	Measures the perceptual similarity between two images. Extracted high-level features from pre-trained networks (e.g. AlexNet [28]) are compared to calculate the degree of similarity.
Pros	Considers high-level semantic features and spatial relationships in images.
Cons	Requires a reference image for comparison - can be challenging diversely generated samples without direct correspondences.

Table 2.3: Learned Perceptual Image Patch Similarity (LPIPS): A quantitative metric for evaluating GANs

Fréchet Inception Distance (FID)	
Description	Measures the similarity between the distributions of real and generated images in the feature space of the pre-trained model (InceptionV3).
Pros	More robust and sensitive to image quality variations than IS, as it compares the statistics of entire distributions, not individual samples.
Cons	High bias: requires huge sample size (often above 50 000). Anything lower will cause the FID to be overestimated the actual value.

Table 2.4: Fréchet Inception Distance (FID): A quantitative metric for evaluating GANs

Manuel Inspection	
Description	Qualitative assessment of generated images by human evaluators, considering factors like realism, sharpness, and content preservation.
Pros	Performs quite well for assessing image quality, as humans have a very natural perception of realism.
Cons	Is subjective and suffers from evaluator bias. It may require domain expertise, and often limited by the number of images that can be examined within a practical timeframe.

Table 2.5: Manuel Inspection: A qualitative metric for evaluating GANs

Employing appropriate evaluation metrics does not guarantee success. Acknowledging that researchers often face challenges or difficulties when using these metrics is essential. Some of these include:

- **Mode collapse**: Evaluation metrics may not adequately detect mode collapse, a phenomenon in which a GAN generates only a limited variety of samples, leading to low diversity in the output.
- **Sensitivity to model choice**: Some metrics, such as FID and IS, rely on specific pre-trained models (e.g., InceptionV3) for evaluation. The choice of the model can affect the evaluation results and may not be relevant to certain domains or tasks.
- **Perceptual inconsistency**: Evaluation metrics may not always align with human perception, potentially leading to discrepancies between quantitative scores and qualitative human evaluations.
- **Subjectivity**: Visual examination, a common qualitative evaluation approach, is inherently subjective and can suffer from evaluator bias, inconsistencies, and scalability issues.
- **Lack of ground truth**: In unsupervised image-to-image translation, there is no single correct output for a given input. The absence of a ground truth complicates the evaluation process and makes defining "good" or "bad" results challenging.
- **Disentangling quality and diversity**: Quantitative evaluation metrics might not effectively disentangle the quality of generated samples from their diversity, making it difficult to assess these aspects independently.

2.3 Image-to-Image Translation

Suppose you have captured a beautiful picture of your dog and wish to turn it into a comic book-style illustration; how can one programmatically use computer-based tools

to achieve this artistic conversion? This kind of research falls under the category of the image-to-image translation [18][41] and has emerged as a significant research area in generative models and deep learning, aiming to convert images from one domain to another while preserving contextual information.

2.3.1 Fundamentals

The primary objective of image-to-image translation is to learn a mapping between distinct image domains, utilizing various techniques to address the challenges associated with diverse image properties and structures. The translation involves learning a mapping function G to convert an input image x from domain X to an output image y in domain Y to minimize the difference between the translated image and the ground truth image. Mathematically, this can be represented as $G : X \rightarrow Y$ with $y = G(x)$.

2.3.2 Paired Image-to-Image Translation

In paired image-to-image translation, the model learns a mapping between two image domains using a dataset containing aligned image pairs, where the input and output images share a one-to-one correspondence.

In the case of paired image-to-image translation, we have a dataset of paired images (x_i, y_i), where $x_i \in X$ and $y_i \in Y$.

2.3.3 Unpaired Image-to-Image Translation

Conversely, unpaired image-to-image translation involves learning a mapping between two image domains without direct correspondence, requiring the model to understand the relationship between the domains without paired training examples.

For unpaired image-to-image translation, one does not have paired samples. Instead, one often uses adversarial training with an additional discriminator network D. The discriminator distinguishes between real samples from domain Y and generated samples from $G(X)$.

2.3.4 Key Architectures and Objective Functions

Several key architectures, showcased in Section 2.2.2, have been developed for both paired and unpaired image-to-image translation, each with unique characteristics. In addition to continuous progression in GAN (generator and discriminator) architecture, more meaningful objective functions have also been developed. For instance, the introduction of cycle-consistency loss by CycleGAN [59], the introduction of conditional adversarial loss by Pix2pix [18], the introduction of style reconstruction loss by StarGAN-v2 [8], the introduction of contrastive learning by combining adversarial loss and PatchNCE loss [42], and the introduction of shared-latent space learning with VAE-GAN [32] have all contributed to the improvement of image-to-image translation quality.

Challenges and Limitations

Despite their advantages, there are challenges and limitations associated with using GANs for image-to-image translation:

- **Training Stability**: GAN training can be unstable, leading to mode collapse, vanishing gradients, and unstable convergence. Techniques like Wasserstein GANs and Spectral Normalization can help address these challenges, as Section 2.2.2 mentions.
- **Evaluation Difficulty**: Evaluating GAN-generated images is challenging due to the lack of ground truth for unsupervised tasks, and the image's quality vs. diversity trade-off complicates the evaluation process.
- **Computational Requirements**: Training GANs, especially for high-resolution image synthesis, can be computationally expensive and usually require significant training data.

2.4 Related Work

The field of unpaired image-to-image translation has picked up academic interest in recent years, with notable developments in pure visual (RGB-to-RGB) translation tasks. Several methods, mostly revolving around models presented in Section 2.2, have demonstrated significant success in unpaired image-to-image translation. However, less work has focused on the more challenging task of visual to "non-visual" (RGB-to-Infrared) translation.

NIR-to-RGB: The study "Colorizing Near Infrared Images Through a Cyclic Adversarial Approach of Unpaired Samples" [33] explores colorizing Near Infrared Images (NIR) using cyclic adversarial methods. Image colorization has been a topic of prior research [37]. This study investigates NIR-to-RGB translation as opposed to RGB-to-IR. The crucial distinction between NIR and IR images is the wavelength range and image type. As illustrated in Figure 2.3, NIR bands span 750 to 2500 nm, whereas the IR cameras utilized in this thesis (see Section 3.1 in Specialization Project [38]) cover 7500 nm – 13500 nm. This range corresponds to the long-wave infrared (LWIR) region.

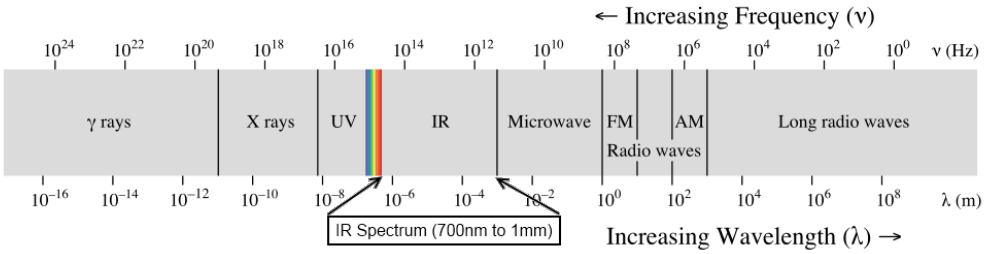


Figure 2.3: Wavelength spectrum, showing the area of the IR range. Image source: https://learn.sparkfun.com/tutorials/light/infrared-light

TIR-to-CV: Another work investigates unpaired thermal infrared image translation [53]. They propose a method called GMA-CycleGAN (Gray Mask Attention-CycleGAN) that translates thermal-infrared images into realistic color visible images. The approach begins with thermal-infrared-to-greyscale-visible translation, followed by greyscale-visible-to-color-visible translation. Although this study investigates thermal infrared image translation, it falls short of providing details on end-to-end inference speed and practical use-cases.

LWIR-to-RGB: Another work investigates unsupervised object detection via long-wavelength infrared (LWIR) to RGB translation [1]. Specifically, they emphasize the goal of creating quality object detectors for IR images using only RGB images for training data. In short, using CycleGAN, they attempt image translation between LWIR and RGB imagery. Instead of creating a specific infrared detector, it seems they translate LWIR images to RGB and then use a trained RGB detector. There are no details regarding end-to-end inference speed or their actual use cases in practice.

Misalignment Issue: "Cross-Modal Alignment Meets RGB-Infrared Vehicle Detection" [54] addresses the misalignment issue between RGB and IR images in the context of multi-spectral aerial detection. Misalignment arises primarily due to position, scale, and rotation deviations when capturing the same scene with two different cameras (RGB and IR), as seen in Figure 2.4.

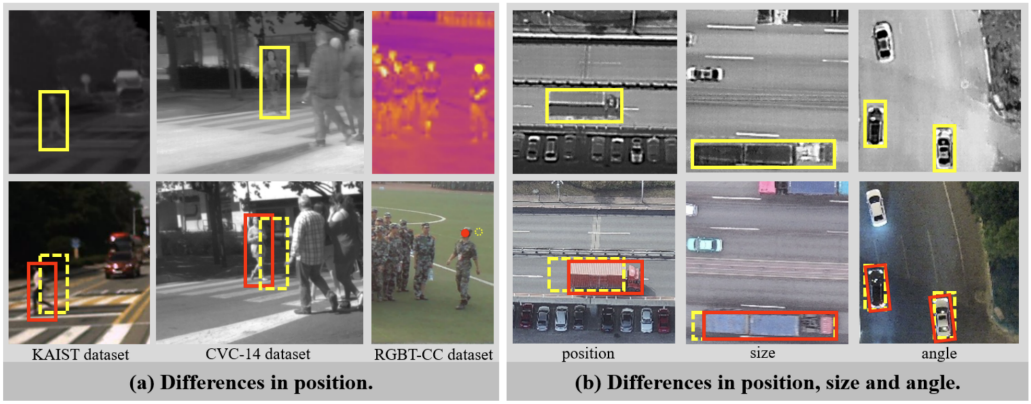


Figure 2.4: A depiction of the misalignment issue in cross-domain translation. Image segments (a) and (b) are visual representations from ground truth annotations. These segments are cropped from the same position of RGB-IR image pairs. Yellow and red boxes signify the annotations of identical objects within infrared and visible images, respectively. Image is sourced from the authors paper [54] with permission.

The field of NIR-to-RGB and other similar domain-specific translation tasks has seen significant progress, yet the literature reveals a slight gap in RGB-to-IR translation. Addressing this gap could provide substantial benefits with the increasing need for cross-domain image translation in various applications such as object detection, remote sensing, and safety detection systems.

This need is primarily fueled by the rising use of autonomous systems across various sectors, including transportation and surveillance. These systems heavily rely on camera technology, and the capability to process and comprehend images across different domains, particularly RGB and infrared, is essential for efficient performance under diverse environmental conditions. Infrared cameras offer a distinct advantage over RGB cameras in certain conditions as they can capture invaluable data in low-light or poor visibility scenarios, thereby enhancing the robustness and reliability of autonomous systems.

However, the major hurdle in this area is the need for paired RGB and infrared datasets that can be used to train robust object detection models. Most existing image datasets are usually RGB and have been captured under normal daylight conditions. While methods for unpaired image translation for RGB-to-RGB or NIR-to-RGB tasks exist, the literature concerning direct RGB-to-IR translation needs further exploration. Specifically, the challenge of utilizing current RGB datasets to boost infrared object detection systems remains unsolved. Hence, exploring generative models for unpaired image-to-image translation from RGB to IR becomes an important research topic to pursue and could mark a significant advancement in autonomous systems.

2.5 Key Concepts

Fundamental concepts and terminologies that establish the groundwork for this project are summarized in the following list:

- **Machine Learning**: A field of study focused on algorithms and statistical models that enable computers to perform tasks without explicit programming. These tasks often involve prediction, classification, or clustering.
- **Deep Learning**: A subset of machine learning that uses deep neural networks with multiple layers to understand complex patterns in datasets. Typical applications include image and speech recognition.
- **Neural Network Components Techniques**: These are the fundamental elements of neural networks like backpropagation, gradient descent, activation functions, and regularization techniques that control the learning process and prevent overfitting.
- **Generative Models**: These machine learning models can generate new data resembling input data. Types include Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and other deep generative models.
- **Generative Adversarial Networks (GAN)**: A generative model consisting of a generator and a discriminator. The generator "generates" realistic data while the discriminator "discriminates" if the data is realistic or fake. The two components are trained jointly in a process that resembles a game, hence the term "adversarial".
- **Image-to-Image Translation**: This involves techniques that use generative models to convert images from one domain to another, like changing day scenes into night scenes or converting sketches into realistic images.
- **Conditional GANs**: GANs that generate data with specific attributes or conditions, often used in tasks such as image synthesis and image-to-image translation.
- **Unpaired Image-to-Image Translation**: A technique often used in GANs where the model learns to translate between two image domains without paired examples, i.e., it can learn to convert horses into zebras from unpaired images of horses and zebras.
- **Paired Image-to-Image Translation**: Unlike unpaired translation, this technique requires paired training examples in both source and target domains. For instance, to convert black and white photos to color, the model would need paired black and white and color images.
- **Cycle Consistency Loss**: A concept often used in unpaired image-to-image translation that enforces consistency between the original and the translated image. If an image is translated from one domain to another and then back to the original

domain, the cycle consistency loss encourages the twice-translated image to match the original image.

- **Evaluation Metrics for GANs**: These are measures used to assess the performance of GANs. Typical metrics include Inception Score (IS), Fréchet Inception Distance (FID), and others. They evaluate aspects like image quality and diversity.
- **Fréchet Inception Distance (FID)**: A evaluation metric that calculates the distance between the distribution of generated images and the distribution of real images in the feature space of a pre-trained Inception Network. Lower FID scores indicate that the two distributions are closer and, therefore, the quality of the generated images is higher.

3 Methods

This chapter outlines the methods and procedures used to address the challenges and goals associated with unpaired image-to-image translation. The chapter begins with a discussion of the data sources utilized, followed by an exploration of the dataset. Next, the selected translation models are presented, along with a description of the evaluation metrics used to assess their performance.

For transparency, some theoretical concepts and equations about CycleGAN and CUT, in Sections 3.3.1 and 3.3.2, have been copied from the Specialization Report [38].

3.1 Data Source

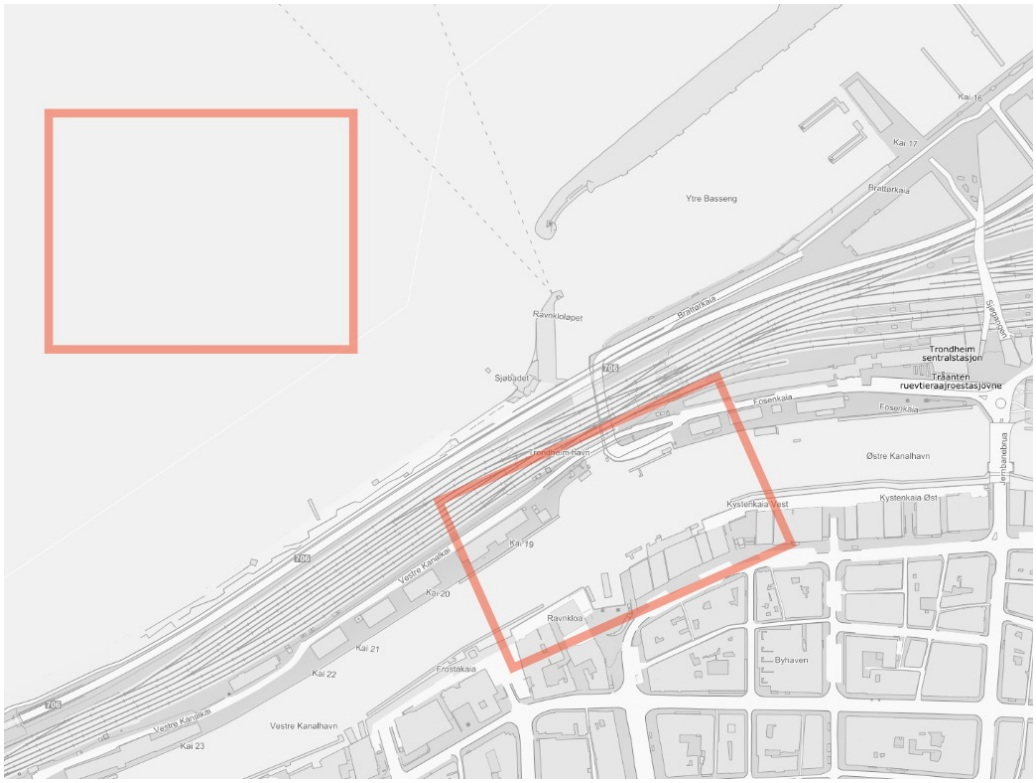


Figure 3.1: Greyscale map of the harbor in Trondheim, Norway. The approximate data collection area is marked in red. Screenshot captured from `https://www.norgeskart.no/`

The images utilized in this thesis stem from special files called rosbags, collected through sea trials conducted by NTNU. As previously stated in the Specialization Project [38], rosbags consisted of various sensor data from the milliAmpere's sensor rig, where

precisely the images are of interest. The raw rosbag images consisted of various driving scenarios, both inside and outside the harbor of Trondheim, recorded by the milliAmpere 1 ferry. The area of data collection is limited to specific boundaries, as seen in Figure 3.1, bounded by two regions outlined in red. The upper region primarily involved stationary data collection, during which various boats were observed traversing the cameras' field of view at medium to far distances (roughly 30-300 m). Conversely, the lower region of the collection area consisted of image captures during stationary positioning and while driving across the channel. The lower area consists of a considerably smaller drivable area than the upper area. The channel's visual context consists of various boats docked at the harbor. As a result, the autonomous ferry was in closer proximity (roughly 5-50 m) to other boats, both docked and in traffic, leading to image captures at much closer distances.

3.2 Data Exploration

This thesis continues with the same camera configuration, stated in Table 3.1 in the Specialization Project [38] and restated as Table 4.1 in this thesis. As previously stated, all raw real-world images are captured using five infrared and five electro-optical cameras mounted on the ferry. Compared to the Specialization Report, where only the *2021-05-05-10-58-01* dataset was used, this thesis now considers all of the readily available data (rosbags) from NTNU - a total of 202 GB. The raw images in the thesis, extracted from each rosbag, consist of 143 117 infrared and 89790 electro-optical images, totaling 232 907 unprocessed images. Post extraction, the images represent a total size of 98 GB as seen in Table 3.1.

RosBag	Image	Optical	Infrared	Size (GB)
2021-05-04-09-54-21	16310	6878	9432	7.0
2021-05-04-10-02-07	12480	5262	7218	5.3
2021-05-04-10-09-41	12799	5397	7402	5.5
2021-05-04-10-17-43	14804	6242	8562	6.4
2021-05-04-10-26-22	18608	7847	1076	8.0
2021-05-05-10-52-10	11901	4383	7518	5.4
2021-05-05-10-58-01	13319	4905	8414	6.1
2021-05-05-11-06-46	8057	2970	5087	3.7
2021-05-05-11-13-36	9941	3661	6280	4.4
2021-05-05-11-34-47	8293	3056	5237	3.8
scenario_1_2021-12-17-10-36-01	10737	3956	6781	4.7
scenario_2_2021-12-17-10-40-17	9387	3458	5929	3.8
scenario_3_2021-12-17-10-44-20	9155	3373	5782	3.7
scenario_4_2021-12-17-10-49-19	12342	4547	7795	4.9
scenario_5_2021-12-17-10-53-56	11735	4322	7413	4.6
scenario_6_2021-12-17-10-58-25	8132	2994	5138	3.2
scenario_7_2021-12-17-11-03-59	6769	2493	4276	2.7
scenario_8_2021-12-17-11-12-56	7813	2875	4938	3.1
scenario_9_2021-12-17-11-16-36	5818	2143	3675	2.3
scenario_10_2021-12-17-11-21-01	6768	2494	4274	2.6
scenario_11_2021-12-17-11-27-31	10670	3930	6740	4.0
scenario_12_2021-12-17-11-33-21	7069	2604	4465	2.7
SUM	232907	89790	143117	97.9

Table 3.1: All of the available RosBags from NTNU

A grid collection of randomly sampled electro-optical images from all rosbags

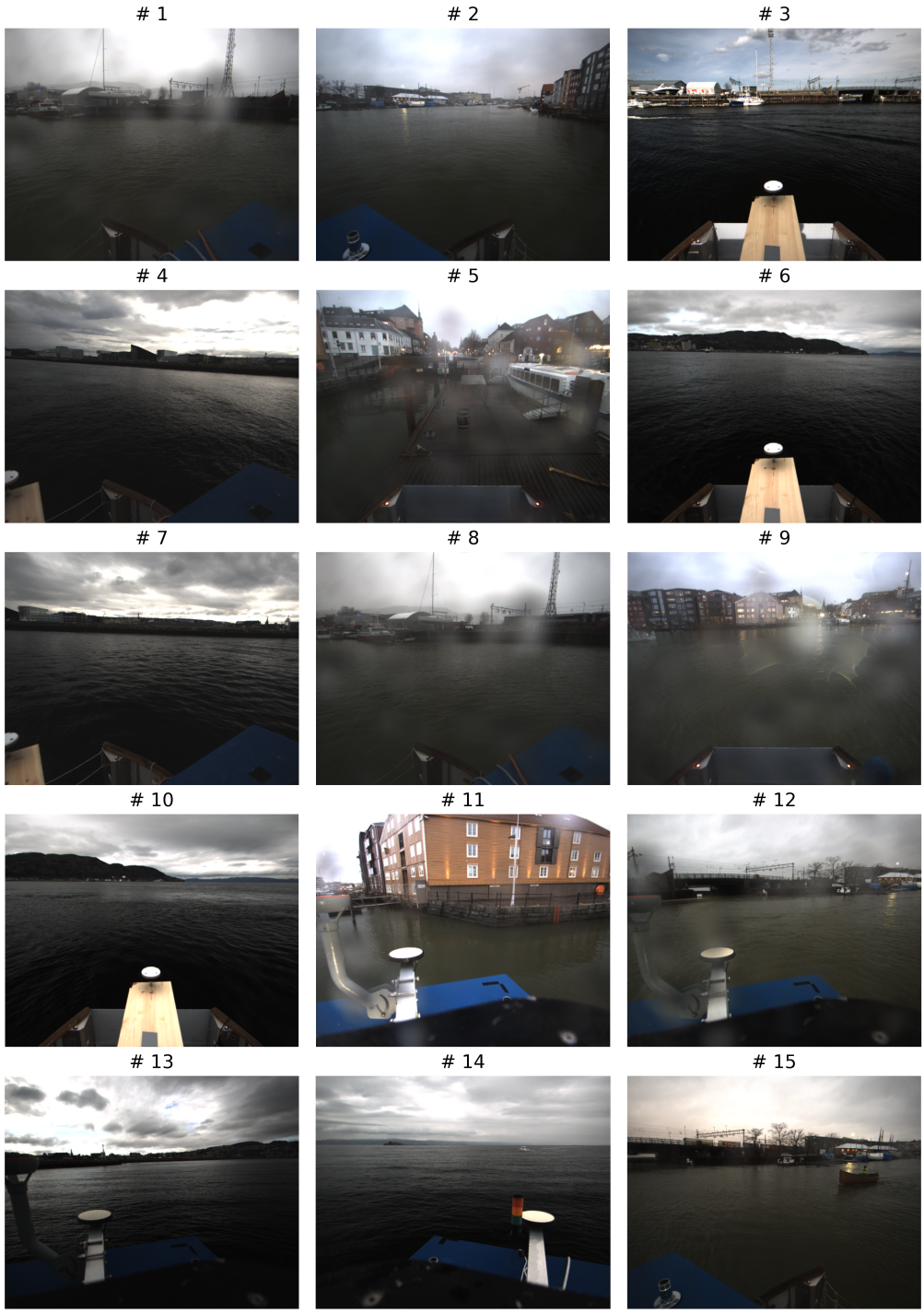


Figure 3.2: 15 randomly sampled optical images from the Rosbags in Table 3.1

A grid collection of randomly sampled infrared images from all rosbags

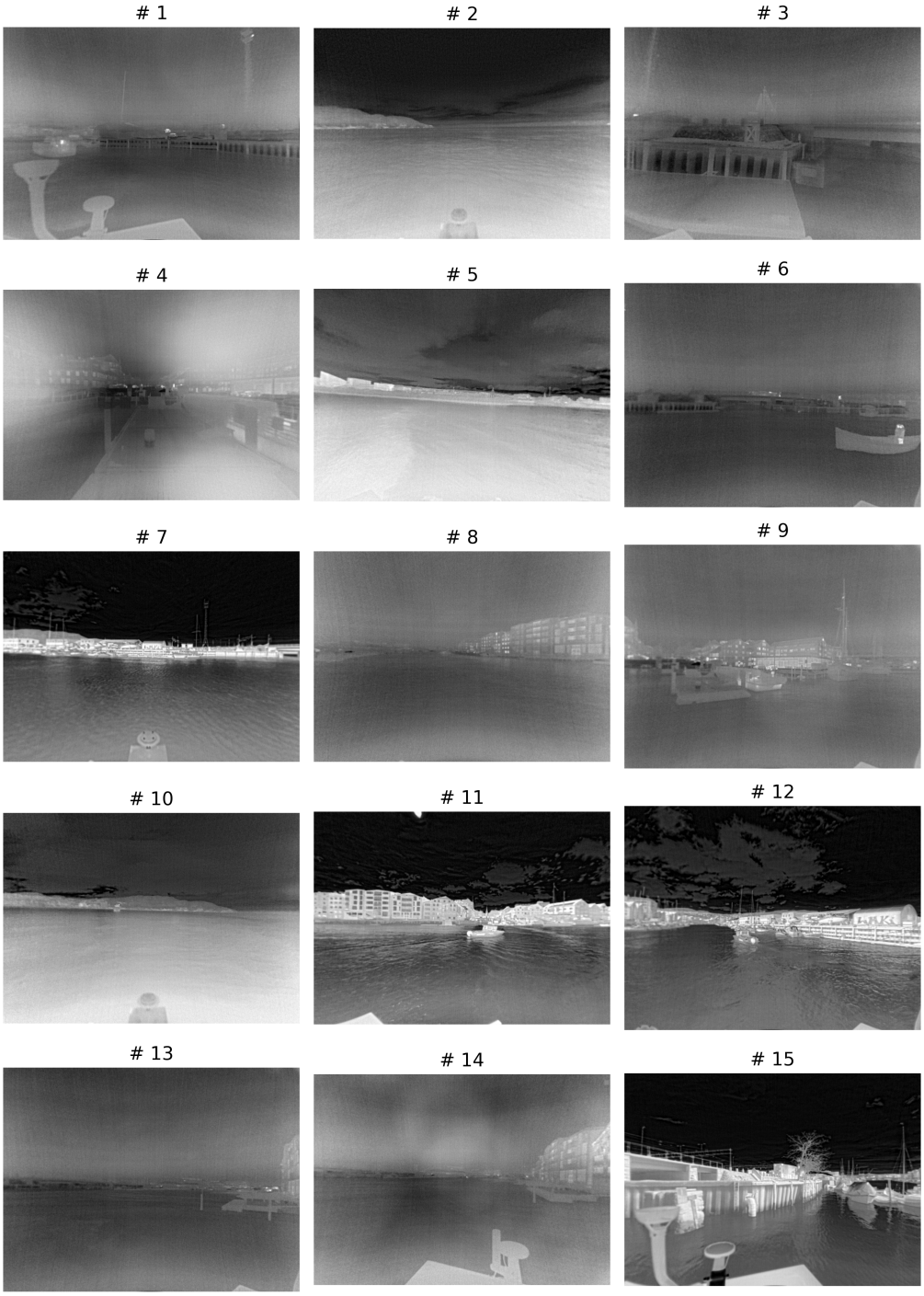


Figure 3.3: 15 randomly sampled infrared images from the Rosbags in Table 3.1

3.2.1 Data Investigation

Upon examining the image collections in Figures 3.2 and 3.3 of electro-optical and infrared images, respectively, it is evident that the occurrence of objects in reasonable proximity to the milliAmpere ferry is a rarity. Both sets of domain images have a small number of images with actual objects of interest in the frame.

Upon further inspecting the electro-optical preview figure, it can be observed that only one out of fifteen images showcases objects of significance. The image indexed at **15** captures a scene of the harbor where only a tiny portion of the image contains the object of interest. In contrast, four out of fifteen images from the infrared collection include interesting objects. Notably, the image indexed at **1** shows the outline of the ferry's pipe, which protrudes in front of the lens, partially obscuring the area immediately beneath it.

It is important to note that several electro-optical images, enumerated as **1**, **2**, **5**, **8**, **9**, **11**, and **12**, exhibit occlusion due to water droplets on the lens. This occlusion results in blurry and unclear images, rendering them unusable for analysis. The same occlusion effect is reflected in the infrared images indexed at **4**, **9**, and **14** (possibly **1** as well), where water droplets are reflected as white zones/spots, further adding to the challenge of utilizing these images for analysis.

With this in mind, the water droplets effect on the lenses poses an additional problem. It renders a substantial number of images almost useless[1]. Consequently, it is necessary to carefully consider data points for analysis to ensure the exclusion of irrelevant or occluded images. It is essential to note that the reduction in the dataset's size leads to a loss of training data. However, to achieve the intended outcomes of the thesis, an extensive and rigorous filtering process is necessary to eliminate irrelevant images to raise the overall quality of the final dataset.

Developing techniques to mitigate the occlusion effect due to environmental factors could improve the quality and utility of the collected data. The process of removing water droplets from images is referred to as image de-raining or rain removal in the research community and has been studied by various researchers [57][55][56]. The task itself is concretely defined as an image-restoration[2] problem, in-which similar to this thesis, uses generative models.

Depending on the camera angle from which the image is taken (five in total), the objects of interest in the lower section of the images (in both domains) may become occluded by the ferry's hull, possibly rendering the image unusable.

Another interesting detail, represented by a smaller amount of images, is the effect of blurring. Visual observation reveals that as the object of interest moves in the opposite direction of the camera, certain objects become slightly blurred or less detailed.

[1] Useless in the context of creating an ideal dataset. The result of creating *ideal lab-conditions* relies on using the best quality data available

[2] The goal of image restoration is to enhance the quality of the degraded image and restore it as closely as possible to its original state, thereby improving its visual appearance and facilitating further analysis or processing

According to Table 4.1, later presented in Section 4.1.1, the difference in the field of view indicated by the supplier, as well as the focal length, indicates that while the electro-optical and infrared cameras capture the same visual scenery, the objects may be represented at different scales and distances. Due to differences in resolution and field of view, only images with objects predominantly in the center of the frame are valuable, as they are more equally represented in both domains. Additionally, the distance between the camera and the object must be considered, as objects may appear significantly smaller in the infrared camera at far distances than in the electro-optical camera. This misalignment between the domains, as shown in Figure 3.4, causes several deviations, such as differences in scale, position, and slight rotation. These deviations are calculated using Euclidean distance and can be found in Appendix C. Specifically, the position (158.41 px) and scale (106.04 px) are significantly different, likely increasing the complexity of image translation between the domains.

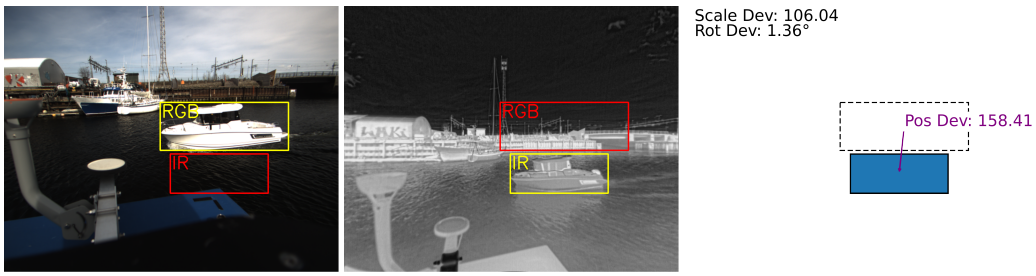


Figure 3.4: Illustration showing the object translation between going from one domain to another. The yellow rectangle indicates the true location and the red rectangle indicates translated location. Note: Both images are approximately taken at the same time. Disclaimer: The IR image has been resized to match the size of the RGB image (from 625x448 to 1122x888) to enhance the visualization.

Various attempts have been made using calibration techniques such as homography to align the two domains correctly without success. Experiments around similar issues (RGB-to-IR) have been researched [29][58], which have revealed extensive techniques to mitigate this such issues. In addition to not having access to any calibrations measure at hand, approaches such as COLMAP[3] could be utilized but requires multiple images of the same scene from different viewpoints.

An important finding related to the infrared domain is that it appears to be represented in two distinct styles, as seen by indices **14** and **15** in Figure 3.3. The first style, visible in index **15**, is characterized by a clear separation between dark and bright grayscale values, typically observed during clear daylight conditions with direct sunlight. In contrast, the second style, visible at index **14**, appears more grainy and exhibits less

[3]COLMAP, an open-source software used for reconstructing the structure of a scene (3D) or object by analyzing multiple images (2D). Uses structure-from-motion and multi-view stereo techniques.

clear separation between dark and bright grayscale values. Cloudy conditions and more uniform temperatures in the scenery will likely cause this style.

In short, the exploration of the raw datasets presents the following challenges:

1. **Limited occurrence** of objects of interest in both electro-optical and infrared image collections, with only a few images containing meaningful objects.
2. **Occlusion due to water droplets** on the lens in several images, resulting in blurry and unclear images, rendering them unusable for analysis.
3. **Object obstruction**: lower section of all the images affected by the structure of the ferry itself, inducing problems when objects of interest become occluded by the ferry's structure.
4. Effect of **blurring** due to moving objects in conjunction with the camera.
5. **Difference in field of view** and camera configuration between electro-optical and infrared cameras, resulting in objects being represented in different scales and distances.
6. **Misalignment** between the two image domains in scale, position, and rotation increases the complexity of the image translation process.
7. **Infrared style varies** depending on weather conditions, resulting in clear or grainy images with varying separation between dark and bright grayscales.

3.3 Model Selection

The subsequent subsections discuss four distinct unpaired image-to-image translation approaches chosen for implementation in this thesis. Analogous to the goal of this project (RGB-IR), these implementations strive to learn a mapping from some domain (A) to another domain (B). Continuous advancements in the unpaired image-to-image translation are often driven by insights accumulated from the foundational CycleGAN architecture. These advancements encompass refinements in objective functions, often inspired by the original adversarial loss [13], network architecture enhancements, and new training techniques.

Numerous models have been considered for selection and evaluation, but some have fallen short for various reasons. Specifically, eligible models should include a functional source implementation code (minor refactors are acceptable). Some interesting research papers on diffusion models have not yet open-sourced their code implementations. While some research papers have been promoting exciting results, they have yet to be further experimented with due to poor code implementation quality. They have been excluded from the process because the time required for refactoring was extreme.

3.3.1 CycleGAN

CycleGAN [59] was chosen because of its previous result in the Specialization Report. Maintaining a reference or baseline to the previous helps to evaluate the progress. CycleGAN employs a cycle consistency constraint that ensures that the learned mappings are consistent from one domain to the other. This attribute of CycleGAN makes it a good candidate for RGB-to-IR image translation tasks, where obtaining perfectly paired matching images is challenging.

The approach presented in their paper does not only learn some mapping G from domain X $\rightarrow$ Y, but the newly established cost function, which essentially focuses on reconstructing the image backward, that is, learning a mapping F : Y $\rightarrow$ X. Using the term, *cycle-consistency*, which adds more structure to the objective function, and has become the standard method for enforcing coherence between domains. The *cycle-consistency* attempts to learn an inverse mapping from the output domain back to the input and checks if the input can be reconstructed.

The full loss objective function consists of the regular adversarial loss in addition to the *cycle-consistency* loss. More specifically, the authors formulate their total objectives as

$$\begin{aligned} \mathcal{L}\left(G, F, D_X, D_Y\right) = & \mathcal{L}_{\text{GAN}}\left(G, D_Y, X, Y\right) \\ & + \mathcal{L}_{\text{GAN}}\left(F, D_X, Y, X\right) \\ & + \lambda \mathcal{L}_{\text{cyc}}(G, F) \end{aligned} \tag{3.1}$$

where specifically they formulate their adversarial loss function $\mathcal{L}_{\text{GAN}}$ similar to (2.1) and the new *cycle-consistency* loss (3.2)

$$\begin{aligned} \mathcal{L}_{\text{cyc}}\left(G, F\right) = & \mathbb{E}_{x \sim p_{\text{data}}(x)}\left[\|F(G(x)) - x\|_1\right] \\ & + \mathbb{E}_{y \sim p_{\text{data}}(y)}\left[\|G(F(y)) - y\|_1\right]. \end{aligned} \tag{3.2}$$

with the notations

- G: generator, maps input images from domain X to domain Y.
- F: generator, maps input images from domain Y to domain X.
- $p_{\text{data}}(x)$: probability distribution of images in domain X.
- $p_{\text{data}}(y)$: probability distribution of images in domain Y.
- $\mathbb{E}x \sim p_{\text{data}}(x)$: expectation over samples x drawn from the probability distribution $p_{\text{data}}(x)$.
- $\mathbb{E}y \sim p_{\text{data}}(y)$: expectation over samples y drawn from the probability distribution $p_{\text{data}}(y)$.
- $||F(G(x)) - x||_1$: L1 norm between the generatd and reconstructed image $F(G(x))$ and the original image x.

- $||G(F(y)) - y||_1$: L1 norm between the reconstructed and generated image $G(F(y))$ and the original image y.

The adversarial loss ensures that the generated images are indistinguishable from the target domain images, while the cycle consistency loss enforces consistent translation between the domains. Together, these two losses help to learn a mapping between the input and output domains.

3.3.2 CUT

Contrastive Unpaired Translation [42] was picked due to being a continuation from the efforts presented in the Specialization Report. Developed by the same group behind CycleGAN, CUT can be considered a successor to that algorithm.

CUT combines adversarial loss with PatchNCE loss, a patch-wise contrastive loss, to encourage consistent and meaningful local image representations. This combination enables the model to preserve local structure and content from the input images.

The objective function consists of the adversarial loss, which ensures that the generated images are indistinguishable from real images in the target domain, and the PatchNCE loss acts as a regularizer to maintain local consistency and minimize the semantic gap between the input and output images. CUT does not use CycleGAN's *cycle-consistency* but instead focuses on maximizing mutual information using a noise contrastive estimation framework. The final objective function can be formalized as (3.3)

$$\mathcal{L}_{\text{GAN}}(G, D, X, Y) + \lambda_X \mathcal{L}_{\text{PatchNCE}}(G, H, X) + \lambda_Y \mathcal{L}_{\text{PatchNCE}}(G, H, Y) \tag{3.3}$$

Denoting the symbols G as the generator, D as the discriminator, and H as a small two-layer MLP network. Besides the images X and Y from their respective domains, the notations can be explained as

- λ_X, regularization term weighted by some scalar
- λ_Y, regularization term weighted by some scalar
- $\mathcal{L}_{\text{GAN}}$, the adversarial loss function
- $\mathcal{L}_{\text{PatchNCE}}$, the noise-contrastive estimation loss function

where specifically they formulate the adversarial loss function as

$$\mathcal{L}_{\text{GAN}}(G, D, X, Y) = \mathbb{E}_{\boldsymbol{y} \sim Y} \log D(\boldsymbol{y}) + \mathbb{E}_{\boldsymbol{x} \sim X} \log(1 - D(G(\boldsymbol{x}))) \tag{3.4}$$

Using this loss, the generator G aims to generate realistic images, while the discriminator D strives to distinguish between real and generated samples. For $\mathcal{L}_{\text{PatchNCE}}$, we have the objective

$$\mathcal{L}_{\text{PatchNCE}}(G, H, X) = \mathbb{E}_{\boldsymbol{x}\sim X} \sum_{l=1}^{L} \sum_{s=1}^{S_l} \ell\left(\hat{\boldsymbol{z}}_l^s, \boldsymbol{z}_l^s, \boldsymbol{z}_l^{S\setminus s}\right) \tag{3.5}$$

which attempts to measures the similarity between corresponding input-output patches and pushes unrelated patches apart in the learned feature space.

- $\mathbb{E}_{\boldsymbol{x}\sim X}$: This is the expected value or mean over all images $\boldsymbol{x}$ drawn from the distribution X. The idea is to calculate the PatchNCE loss over all images in the source domain X.
- $\ell\left(\hat{\boldsymbol{z}}_l^s, \boldsymbol{z}_l^s, \boldsymbol{z}_l^{S\setminus s}\right)$: This is a loss function ℓ that takes as input three arguments. $\hat{\boldsymbol{z}}_l^s$ represents the predicted or generated output at the l-th layer and s-th patch. $\boldsymbol{z}_l^s$ is the actual output at the l-th layer and s-th patch, and $\boldsymbol{z}_l^{S\setminus s}$ represents all other outputs at the l-th layer excluding the s-th patch.
- $\boldsymbol{z}$ and $\hat{\boldsymbol{z}}$: These are vector representations of the patches in the feature space at a specific layer.

3.3.3 GcGAN

The reason for selecting GcGAN [12] was primarily due to its exciting take on its geometry-consistency constraints for one-sided unsupervised domain mapping. The idea of preserving the geometry of a scene during translation is essential. The constraints consist of taking the original image and its geometrically transformed counterpart as inputs and generating two images in the new domain while maintaining geometry consistency.

GcGAN sets itself apart from other models by utilizing geometry consistency constraints. This property ensures that fundamental geometric transformations do not change the semantic structure of the images. As a result, GcGAN claims to prevent mode collapse, a common issue with standard GANs, by effectively translating networks on original and transformed images via co-regularization. This approach also helps to reduce semantic distortions during translation.

The full loss objective function for GcGAN is given in Eq. ((3.6)):

$$\begin{aligned}\mathcal{L}_{GcGAN}\left(G_{XY}, G_{\tilde{X}\tilde{Y}}, D_Y, D_{\tilde{Y}}, X, Y\right) =& \mathcal{L}_{\text{gan}}\left(G_{XY}, D_Y, X, Y\right) \\ &+ \mathcal{L}_{gan}\left(G_{\tilde{X}\tilde{Y}}, D_{\tilde{Y}}, X, Y\right) \\ &+ \lambda \mathcal{L}_{geo}\left(G_{XY}, G_{\tilde{X}\tilde{Y}}, X, Y\right)\end{aligned} \tag{3.6}$$

composed of the three following terms. The first two, being variations of the adversarial loss (3.7). Using the first term as an example, the formulation is as follows

$$\begin{aligned}\mathcal{L}_{gan}\left(G_{XY}, D_Y, X, Y\right) =& \mathbb{E}_{y\sim P_Y}\left[\log D_Y(y)\right] \\ &+ \mathbb{E}_{x\sim P_X}\left[\log\left(1 - D_Y\left(G_{XY}(x)\right)\right)\right]\end{aligned} \tag{3.7}$$

- G_{XY}: The generator model that maps input images from domain X to domain Y.
- D_Y: The discriminator model that distinguishes between real images from domain Y and generated images from domain X.
- X: The set of images in domain X.
- Y: The set of images in domain Y.
- P_X: The probability distribution of images in domain X.
- P_Y: The probability distribution of images in domain Y.
- $\mathbb{E}y \sim P_Y$: The expectation over samples y drawn from the probability distribution P_Y.
- $\mathbb{E}x \sim P_X$: The expectation over samples x drawn from the probability distribution P_X.

This term represents the standard generative adversarial loss for the generators (G_{XY} and $G_{\tilde{X}\tilde{Y}}$) and discriminators (D_Y and $D_{\tilde{Y}}$) networks, considering the original (X, Y) and translated ($\tilde{X}$ and $\tilde{Y}$) image respectively. Lastly, the geometry-consistent objective (3.8), formulated as

$$\begin{aligned}\mathcal{L}_{geo}\left(G_{XY}, G_{\tilde{X}\tilde{Y}}, X, Y\right) =& \mathbb{E}_{x\sim P_X}\left[\left\|G_{XY}(x) - f^{-1}\left(G_{\tilde{X}\tilde{Y}}(f(x))\right)\right\|_1\right] \\ &+ \mathbb{E}_{x\sim P_X}\left[\left\|G_{\tilde{X}\tilde{Y}}(f(x)) - f\left(G_{XY}(x)\right)\right\|_1\right].\end{aligned} \tag{3.8}$$

enforces the geometry-consistency constraint between the generated images by both generators (G_{XY} and $G_{\tilde{X}\tilde{Y}}$) in the new domain. The constant λ is a hyperparameter that regulates the importance of this constraint.

3.3.4 StarGAN-v2

StarGAN-v2 [8] was chosen primarily due to its promising image quality results and multi-domain translation capabilities. In addition to showing SOTA results for various evaluation metrics, the idea of investigating whether similar approaches would work for translating images from the visible domain to the infrared was decided. StarGAN-v2 uses various tricks to help the learning process, such as alignment models to assert that each image is re-positioned so that the eye sockets are precisely at the same pixel position for all images. Not directly transferable to the data in this thesis, as aligning objects is an inherently tricky challenge as discovered during the data investigation presented in Section 3.2.1, it was still an exciting idea open to be explored.

What makes StarGAN-v2 unique compared to other models is its ability to generate output images with various styles across different domains, outperforming previous state-of-the-art methods. It utilizes a single framework to address diversity and scalability challenges, common issues in image-to-image translation models. StarGAN-v2 utilizes a multi-task learning setup and an adaptive instance normalization technique that injects style codes into the generator.

The objective function can be expressed as follows:

$$\begin{aligned}\min_{G,F,E}\max_{D} &\mathcal{L}_{adv} + \lambda_{sty}\mathcal{L}_{sty} \\ &- \lambda_{ds}\mathcal{L}_{ds} + \lambda_{cyc}\mathcal{L}_{cyc}\end{aligned} \tag{3.9}$$

which consist of several components weighted by various lambda parameters. Firstly, an adversarial objective (3.10), similar to what has been discussed previously.

$$\begin{aligned}\mathcal{L}_{adv} =&\mathbb{E}_{\mathbf{x},y}\left[\log D_y(\mathbf{x})\right] + \\ &\mathbb{E}_{\mathbf{x},\widetilde{y},\mathbf{z}}\left[\log\left(1 - D_{\widetilde{y}}(G(\mathbf{x},\widetilde{\mathbf{s}}))\right)\right]\end{aligned} \tag{3.10}$$

Secondly, an reconstruction objective (3.11) which encourages the generator to utilize the style code when generating the output image.

$$\mathcal{L}_{sty} = \mathbb{E}_{\mathbf{x},\widetilde{y},\mathbf{z}}\left[\left\|\widetilde{\mathbf{s}} - E_{\widetilde{y}}(G(\mathbf{x},\widetilde{\mathbf{s}}))\right\|_1\right] \tag{3.11}$$

Thirdly, a style diversification objective (3.12) which attempts to regularizes the generator to create diverse output images by minimizing the similarity between outputs given different style codes.

$$\mathcal{L}_{ds} = \mathbb{E}_{\mathbf{x},\widetilde{y},\mathbf{z}_1,\mathbf{z}_2}\left[\|G\left(\mathbf{x},\widetilde{\mathbf{s}}_1\right) - G\left(\mathbf{x},\widetilde{\mathbf{s}}_2\right)\|_1\right], \tag{3.12}$$

Fourthly, it has a preserving-source-characteristics objective (3.13) which ensures that the generator maintains domain-invariant characteristics of the input image. This could for instance be pose, in the output image by employing a cycle-consistency loss.

$$\mathcal{L}_{cyc} = \mathbb{E}_{\mathbf{x},y,\widetilde{y},\mathbf{z}}\left[\|\mathbf{x} - G(G(\mathbf{x},\widetilde{\mathbf{s}}),\hat{\mathbf{s}})\|_1\right] \tag{3.13}$$

In short, these four objective functions collaborate to ensure that the StarGAN-v2 model effectively generates diverse images across multiple domains. Without repeating similar notations used in previous subsections, the following symbols can be further explained:

- $\mathbf{z}$: This is typically a random noise vector input to the generator in a generative adversarial network (GAN). The randomness of $\mathbf{z}$ helps to generate diverse output images.

- $\widetilde{\mathbf{s}}$: This represents a style code used to control the style of the output image in StarGAN-v2. The style code $\widetilde{\mathbf{s}}$ is typically sampled from a distribution (hence the tilde).
- $\mathbf{s}_1, \mathbf{s}_2$: These represent different style codes. In the context of the style diversification loss, $\mathbf{s}_1$ and $\mathbf{s}_2$ are different style codes, and the goal is to encourage the generator to produce diverse output images when given these different style codes.

3.4 Evaluation Metrics

This thesis uses two evaluation metrics to measure the performance of unpaired image-to-image translation models. As mentioned in Section 2.2.3, many different quantitative metrics exist for generative models. For simplicity, the thesis uses one quantitative and one qualitative evaluation metric. Using the Fréchet Inception Distance (FID) and Manual Inspection (visual observation), common in research papers for GANs, the results become more understandable and easier to compare with other research experiments.

3.4.1 Fréchet Inception Distance

The FID metric provides a reliable assessment[4] of the generated images' quality and diversity have been shown to correlate well with human judgment. It measures the statistical similarity between the generated and target images by comparing their feature distributions in the InceptionV3 [51] network's intermediate layer. It offers an objective way to compare different models and identify which generates images closer to the target distribution. For image-to-image translation models, FID has become an industry-standard evaluation technique due to its ability to capture the diversity and quality of the generated images. Not all metrics are perfect, and some sensitivity issues exist as previously mentioned in Section 3.8.1 in the Specialization Project [38].

FID was chosen over other metrics because it is widely used and well-established in the research community for GANs. Using industry-standard evaluation metrics naturally makes comparing with equivalent research experiments easier, leading to a more comprehensive understanding of the problem.

3.4.2 Visual Observation

Visual observation is a qualitative evaluation method that relies on manually inspecting the generated images to assess their quality and similarity to the target images. This evaluation method considers several factors, such as realism, consistency, detail preservation, sharpness, and the presence of artifacts. While visual observation is subjective

[4]Regarding "reliable assumptions", if the real images are complex or have high variability in their features, it can be difficult for the generator to capture the underlying distribution, leading to a higher FID score. Hence, it is essential to consider the complexity and variability of the real images when interpreting FID scores.

and can vary between evaluators, it is essential to determine the real-world applicability and usefulness of the generated images.

Evaluation Scheme

A visual evaluation scheme has been developed to assess generative models' performance qualitatively. This evaluation scheme is based upon the principles of the Likert-scale [31]. In short, the scheme relies on the criteria: realism, consistency, detail preservation, sharpness, and the presence of artifacts. By assigning a score between 1 and 5 for each criterion, one can compare the models and identify their strengths and weaknesses. Table 3.2 presents the visual evaluation scheme used in this thesis:

Criteria	**Weight**	**Score (1-5)**	
Realism	0.25	1: Highly unrealistic	5: Highly realistic
Consistency	0.20	1: Highly inconsistent	5: Highly consistent
Detail Preservation	0.20	1: Poor detail preservation	5: Excellent detail preservation
Sharpness	0.20	1: Highly unsharp	5: Highly sharp
Artifacts	0.15	1: Numerous artifacts	5: Minimal artifacts

Table 3.2: Visual Evaluation Scheme for Generative Image Translation Models

Each criterion is weighted by its importance. Even though the nuances between them are subtle, the justification for each weight is described as follows:

- **Realism (0.25)**: Models generate synthetic data, hence achieving realistic translations with appropriate heat signatures is the main contributing factor to a well-translated image.
- **Consistency (0.20)**: The translation between the two domains involves a resolution and field of view change. Consistently maintaining these aspects is essential for a well-translated image. It is less critical than Realism, so a slightly lower weight is assigned.
- **Detail Preservation (0.20)**: Details may carry important information about the object's properties. Translation from a high-resolution image to a lower one reduces image quality - hence loss of details.
- **Sharpness (0.20)**: Balancing the Sharpness between the domains may hinder identifying specific details or patterns in the image provided, especially with challenging images affected by water droplets or blurring effects. However, Sharpness is slightly less important than Realism, as the latter encompasses the overall appearance of the generated image.
- **Artifacts (0.15)**: Even though minimizing artifacts is an essential factor of quality translated images, it is considered somewhat less important than the other criteria since certain artifacts may not hinder the identification or interpretation of objects.

To calculate the final score for each model, the weighted average method is used for each criterion and their respective weights. In short, the final score can be calculated as

$$\text{Final Score} = \sum_{i=1}^{5} (W_i \times S_i)$$

where W_i denotes the weight for criteria i, and S_i denotes the score for criteria i.

To enhance the readers interpretation of the visual scheme, a few examples is provided below in Figure 3.5.

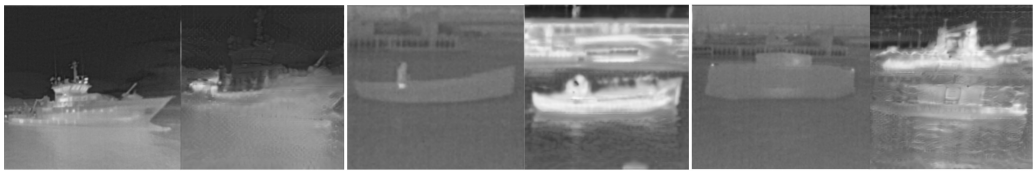


Figure 3.5: Illustration showing the three different images generations: from reference images to into generated images.

Without evaluating in-depth, one can note the following discrepancies in the figure. Observing case (a), a translation of a ferry, one can note that the heat signatures are misplaced comparing the reference and generated images, contributing to a lower realism score - even though it uses the correct grayscale color combinations. In case (b), detail preservation is lacking throughout the translation. It incorrectly introduces some new details not found in the reference image, such as the wave reflection from the surface in front of the boat. Observing case (c), the translation of the research ferry lacks sharpness, resulting in a more blurred expression than the reference. Also, similar to (b), the translation does not correctly preserve the correct details, as one can note the NTNU logo's outline on the hull's side. This suggests various issues, such as incorrect heat signatures, as the logo would disappear in true infrared.

4 Implementation

This chapter reviews the application of the methods and techniques presented in the previous chapter. It enlightens the process of creating a new, customized dataset tailored to the thesis's specific research goals and offers insights into the employed hardware configuration and training pipeline. The experimental setup section outlines the dataset split, pre-processing techniques, and training details for the investigated models. In addition, the evaluation details section highlights the quantitative and qualitative metrics used to conduct the performance of the models on the unpaired image translation task.

For transparency, Table 4.1 is copied from the Specialization Report [38].

4.1 Creating a new Dataset

Deep learning models' efficacy often depends on the datasets employed. Models learn patterns, distributions, and relationships in the data, so if the dataset is biased, unbalanced, or not representative of the real-world problem, it can lead to poor model performance and generalization. In other words, the success of deep learning models is substantially reliant on data quality. However, an exception to the data quantity requirement exists for pre-trained models. In such cases, specific partitions of the pre-trained model may be trained, given the model's previous training on other datasets.

Previous Dataset

In the previous Specialization Project, the dataset consisted of full-resolution images. This resulted in inferior translation performance (see Section 5.3 and 5.4 in the report [38]), and thus, a new dataset is proposed using the approach of only object crops. There are reasons to believe this new approach will result in better performance.

Advancements

Firstly, by employing object crops, the focus is placed on specific objects of interest, effectively reducing image complexity and ensuring that images are more concentrated on the desired subject matter. This eliminates much of the background information in the dataset, such as houses and open waters, streamlining the training process.

Secondly, using object crops allows the capture of fine-grained details of the objects, which is crucial for learning a realistic quality translation between the two visual domains.

Thirdly, object crops provide a consistent and standardized view of the objects, which helps reduce issues such as different lighting, unfortunate camera angles, and perspectives in full-size raw images.

Reducing full-resolution images down to image crops involves an image annotation process, often a time-consuming process, especially when dealing with large datasets. By utilizing a dataset consisting of object crops, models can be trained to focus on smaller objects instead of a high-dimensional image, resulting in better accuracy and more targeted output. Consequently, the additional effort required for image annotation and dataset creation may pay off in terms of performance and realistic quality image translation.

4.1.1 Preparing the Dataset

The data-preparation pipeline for attempting to create a high-quality, customized dataset requires extensive image processing and filtering. Specifically, the steps needed to condense the enormous 98 GB (see Table 3.1) of raw images into useful image crops require the following steps:

1. RosBag to Image
2. Image Filtering
3. Image Downsampling
4. Image Annotation
5. Image Cropping
6. Image Removal

Step 1: RosBag to Image

The initial step in the data pipeline is to convert the data from RosBag[1] files to individual images. This process involves the extraction of the sensor data from the RosBags using the CvBridge[2] from ROS (Robot Operating System), followed by the conversion of the sensor data into individual images. The images are then saved in their respective directories for further processing. This step is identical to the similar process in the Specialization Project. For more details about extracting and undistorting the raw images, please see Section 4.1.1 from the report [38].

[1] A RosBags file is a binary file that contains a collection of ROS messages that can be played back in a ROS system at a later time. It can be used for recording and playback of messages, including sensor data, robot state information, and other data exchanged between ROS nodes.

[2] CvBridge is a software library in ROS that provides a bridge between ROS messages and OpenCV images

Step 2: Image Filtering

Due to the challenges presented in Section 3.2.1, a crucial step in the data pipeline is image filtering. This step involves removing all irrelevant images, images occluded by water droplets, or even the ferry's structure. The filtering process is a crucial step in the data pipeline, as it involves careful visual inspection and selection of each image to ensure that only those containing objects of interest are retained in the data pipeline. For this thesis, this step resulted in the removal of the majority of available data, as most driving scenarios consisted of sequential image series where both domain cameras were active at all times, regardless of the presence of objects. This resulted in long series of images where objects such as boats, ships, and kayaks were only present at specific time intervals. The inspection also shows many duplicate infrared images, most likely due to the reduced video rate legislated by the U.S. government. As FLIR states themselves:

> "Thermal cameras operating at 60 fps and/or 30 fps (NTSC) or 50 / 25 fps (PAL) video rates are export-controlled by the U.S. government."

[3]. The duplication of infrared images is likely a result of trying to produce the same number of images despite the reduced frame rate. A hash-based approach can be used to remove duplicate images from a dataset. The following code snippet involves computing hash values for each image, comparing them to identify duplicates, and then removing them.

```python
import os
import hashlib
from PIL import Image

def remove_duplicates(directory: str) -> None:
    hash_dict = {}
    for subdir, _, files in os.walk(directory):
        for file in files:
            full_path = os.path.join(subdir, file)
            with Image.open(full_path) as img:
                file_hash = hashlib.md5(img.tobytes()).hexdigest()
            if file_hash in hash_dict:
                os.remove(full_path)
            else:
                hash_dict[file_hash] = full_path
```

Step 3: Image Downsampling

Data downsampling is necessary because the electro-optical and infrared images have been sampled at frequencies 5 and 9 FPS, respectively. As a result, there are more

[3] https://www.flir.com/support-center/oem/as-far-as-30-fps-vs.-9-fps-video-rates-are-concerned-why-use-one-over-the-other/

infrared images than electro-optical images. To address the sampling issue, the scenery of each image series's first and last frame should be identical to retain the same amount of information in both domains, and any "extra" images should be removed uniformly in the excess domain. Moreover, a few images might have negligible movement or standstill, rendering them redundant. Depending on the degree of movement in the view anywhere between 50% (i.e., every other image) to 75% (i.e., three subsequent images) of the images can be removed. This will guarantee that the final dataset has an equal representation of the electro-optical and infrared domains while including only the most relevant images. Including numerous images from a specific scene may accidentally introduce an undesirable bias in the overall training data.

Step 4: Image Annotation

After filtering the images, the next step is image annotation. Annotation involves manually creating bounding boxes around objects of interest in the images. As mentioned, this thesis focuses on objects of particular interest instead of the whole image, as the Specialization Project did. This involves objects that are important to detect for the ferry's operation, such as other vessels, swimmers, and kayaks. The annotation process was done using a specialized annotation tool, LabelImg[4], where each object was centered inside a bounding box.

Step 5: Image Cropping

After the annotation process, the next step involves generating image crops using the XML[5] annotation files generated by LabelImg as reference. The purpose of cropping is to extract only the object of interest from the original image without too much scenery background. This process involves using the bounding box annotations to generate new images containing the object of interest. Subsequently, one original image can turn into multiple image crops, each containing a specific object of interest.

Step 6: Image Removal

Following image cropping, a two-stage removal process was implemented. Stage **A** involved filtering out low-resolution image crops, while stage **B** focused on removing excess domain images.

A: When cropping images, bounding boxes can be created in all types of resolution. In this case, many image crops, particularly from the infrared domain, resulted in low resolution due to the differences in the field of view and resolution between the RGB and IR cameras. Due to the different sensor sizes and focal lengths for these cameras, as

[4] https://github.com/heartexlabs/labelImg

[5] XML stands for eXtensible Markup Language and is a markup language used for structuring and storing data in a format that is both human-readable and machine-readable.

stated in Table 4.1, electro-optical objects are larger than their corresponding infrared versions, particularly at far distances.

Specs	**FLIR BlackFly 2**	**Flir Boson 640**
Sensor	electro-optical	infrared
Type of lens	Kowa LM6JC	Kowa LM6JC
Focal length	6 mm	4.9 mm
Sensor size	2/3	2/3
Max resolution	2448 x 2048	640 x 512
HFOV according to supplier	81.9 deg	95 deg
Interface	GigE	USB 2.0
Max framerate	22 FPS	9 FPS

Table 4.1: Hardware specification for both of the cameras used on the milliAmpere ferry.

Consequently, bounding boxes can become quite small, resulting in image crops that need more pixel details to determine if an object is present accurately. In such situations, removing the image entirely from the dataset is more appropriate, as it is not possible to distinguish between noisy objects and small class objects. To this end, a minimum width and height threshold of 30 pixels was established, and all square images below this threshold were removed from the dataset. This approach was adopted to maintain the quality and accuracy of the dataset, particularly in cases where low-resolution images could impact the performance of the deep learning models.

B: One ensures that the electro-optical and infrared domains are equally represented in the dataset by removing excess images from the larger domain. This step involves counting the number of images in each domain and removing the excess images from the larger domain until the number of images in each domain is equal. This process ensures that the dataset is balanced and that the deep learning models trained on the dataset are not biased towards any particular domain. The filtering method consisted of sorting the excess domain images by resolution and removing the lower N images, where N is the difference in the number of images between the domain. In practice, this processing step removed numerous infrared images as they were tiny compared to their approximate electro-optical equivalent. This removal led, again, to the removal of electro-optical images to balance the counts.

4.1.2 Final Dataset

The final dataset was created by following the steps detailed in Section 4.1.1 to ensure that the resulting dataset is of higher quality and not affected by the issues highlighted in Section 3.2.1.

The condensed dataset consists of 3551 image crops from the optical and infrared domains in various sizes. As shown in Table 4.2, the average resolution of the image

crops from the optical domain is approximately $200x200$ pixels, while in Table 4.3, the average resolution of the image crops from the infrared domain is approximately $100x100$ pixels.

A significant effort was made to ensure that both cameras captured the same visual content. However, due to the differences in the field of view and the depth information for each camera, perfect pixel-to-pixel matches per object could only be achieved in some cases. In addition, the raw mean width and height for both domains, at 1102x872 and 619x443, respectively, mean that objects are represented differently in each domain, making unpaired image translation across resolutions more challenging.

	count	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**
raw width	3551	1102	8	1094	1096	1096	1110	1122
raw height	3551	872	6	867	869	869	877	888
crop-width	3551	207	104	92	133	174	248	672
crop-height	3551	206	102	91	132	174	248	665

Table 4.2: Electro-optical image statistics

	count	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**
raw width	3551	619	2	617	617	620	621	625
raw height	3551	443	8	438	438	441	445	468
crop-width	3551	104	51	30	67	90	130	337
crop-height	3551	104	51	30	67	90	130	336

Table 4.3: Infrared image statistics

In order to efficiently investigate the object distributions of the dataset, a web application and application programming interface was developed as a part of this thesis for the purpose of manually classifying images. Attempts to use pre-trained image classifiers were made but without sufficient success. This approach was taken due to the difficulty in skimming through 3551 electro-optical images in a folder using a standard file explorer. The web application was developed using Python libraries such as FastAPI, Uvicorn, and TinyDB for handling the backend services. At the same time, the user interface was created with React and Ant Design for a simplistic and elegant front-end experience. See more details in Appendix D. This greatly facilitated the manual classification process and allowed storing of a JSON database containing images and their respective class labels.

As a result of this approach, a significant amount of effort was dedicated to manually classifying all 3551 electro-optical images in the dataset. Preemptively, the investigation assumed similar characteristics in both domains; hence, only one was explored. The class investigation revealed the following properties, as visualized in Figure 4.1.

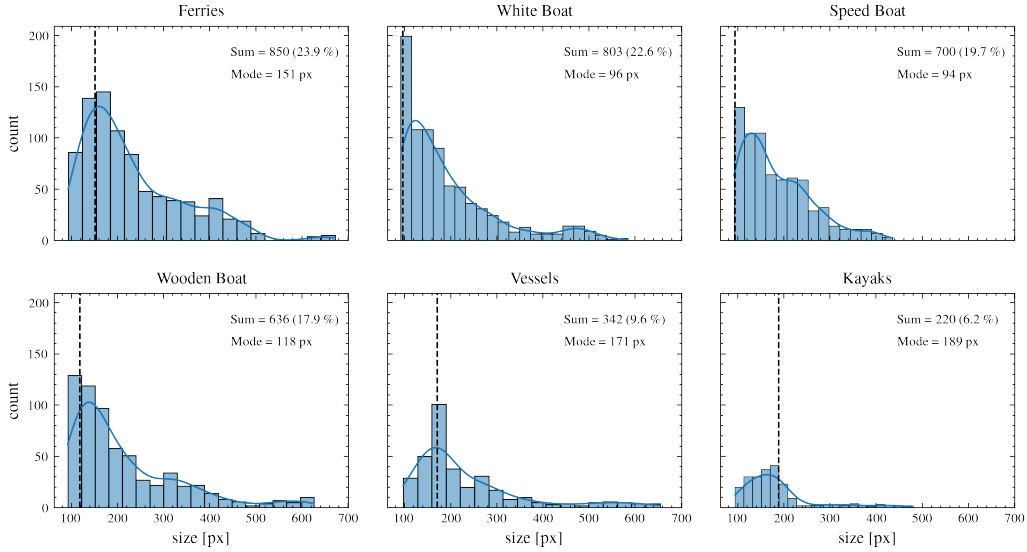


Figure 4.1: The multi-plot shows the distribution of object images in the dataset. Each subplot shows the histogram and kernel-density-estimate contour for each object. The "Sum" in each cell (top-right corner) indicates the total number of image objects, while the percentage indicates its proportion of the whole dataset. Note: image crops are square; hence size = width = height.

Furthermore, the figure reveals the following observations:

- **Class Imbalance and Diversity**: The dataset exhibits a certain degree of class imbalance, with the number of samples for each class varying notably. For instance, ferries and white boats account for almost 47% of the total images, while kayaks represent just over 6%.
- **Volume and Complexity**: The dataset comprises images of varying resolutions with different complexities. The variability in resolution and complexity can pose challenges during training, as the model must learn to adapt to different image sizes and object details. The dataset contains 3,551 images per domain, which may be considered a moderate-sized dataset for training deep learning models. However, the limited volume for specific classes, such as kayaks, may affect the model's ability to effectively learn the features of these less-represented classes. By comparing the class imbalance and the mode of each class, one can observe that, even though images of speed boats are more frequent than kayaks, the mode of kayaks is higher, indicating a potentially higher quality of image translation for kayaks.
- **Resolution and Distance**: By comparing the distribution of each respective object, one can observe significant variations in pixel resolutions. A smaller range of pixel resolutions may indicate images taken at a consistent distance from the boat.

In comparison, a more extensive range of pixel resolutions may indicate images taken at varying distances from the boat. This may affect the quality of the image translations, as the model may not be able to learn the features of the objects effectively. Infrared images have lower resolution than electro-optical images. This could make it challenging to generate high-quality infrared images that closely resemble the corresponding optical images. To address this, one could investigate advanced approaches such as applying super-resolution [30] to enhance the spatial resolution of an image. These techniques aim to reconstruct a higher-resolution image from one or more low-resolution versions of the same image.

4.2 Hardware and Software

Hardware configuration is crucial for practical training of deep learning models, with access to high-end Graphics Processing Units being a determining factor in ensuring the use of computation resources necessary for efficient training.

A remote computer at NTNU with three NVIDIA GeForce RTX 3090s was utilized to conduct model development and training experiments in this thesis. Access to powerful hardware is highly beneficial for more straightforward experimentation setup, training, and testing, with its impact being most pronounced in the actual possibility of training large generative image models. Such models are computationally intensive and require significant resources and time to train accurately.

Compared to cloud-based services like Paperspace[6], access to local resources via NTNU significantly reduced the overall experimentation time without the complexity and cost of using cloud providers that require subscription payments.

In terms of software implementations, the model implementations, training, and testing processes were carried out using the Python[7] programming languages with PyTorch [43], an open-source machine learning library that supports comprehensive tensor computations and deep learning support.

4.3 Training Pipeline

The training pipeline for this thesis involves the remote PC described earlier. To test models and architectures, one can connect to the remote PC via SSH while connected to the NTNU network. If not connected to the network, one can use NTNU's VPN to log into the school intranet and reach the remote PC via SSH.

After cloning the desired code repository, Visual Studio Code[8] was used to configure the training parameters, set up the correct training configurations, and validate the preprocessing techniques.

[6] https://www.paperspace.com/
[7] https://www.python.org/
[8] https://code.visualstudio.com/

The experimentation and logging tracking are performed using an ML Ops platform called Weights and Biases[9]. Weights and Biases (W&B) is a platform that provides researchers with tools to manage and track their machine learning experiments, visualize results, and collaborate with their teams. Once the necessary setup is complete, a training script is executed inside a terminal multiplexer, which supports background processing. This enables the user to detach (exit) the SSH connection without interrupting any ongoing training processes on the computer.

W&B provides various tools for tracking training processes in the cloud. For example, receive alerts (via E-mail or Slack) when training sessions crash, view metrics, and monitor ongoing image generations. This setup enables continuous monitoring of the model's performance and makes the overall training process quite structured.

4.4 Experimental Setup

This section outlines the dataset split, pre-processing, and training details employed to train the various models. See Appendix A for references to detailed code implementations.

4.4.1 Dataset

In order to achieve the objective of obtaining models with satisfactory performance and good generalization capabilities to previously unseen data, the study employs a train/test split methodology. The dataset is divided into two subsets: training and testing sets. The division follows an 80/20 ratio, with 80 percent of the dataset used for training and 20 percent for testing.

The customized dataset, which consists of 3551 images in each domain, is divided into the following splits:

- The RGB training set = 2840 images.
- The RGB testing set = 711 images.
- The IR training set = 2840 images.
- The IR testing set = 711 images.

The purpose of the train/test split is to evaluate the chosen models' performance on unseen data to evaluate generalizability. Using a separate testing set, one can obtain an unbiased estimate of the model's performance and ensure that it can generalize well to new data. All subsequent models use this train/test split.

[9] https://docs.wandb.ai/

4.4.2 Training Details

Throughout the course of this semester, as part of the thesis, a considerable amount of time was dedicated to implementing and experimenting with various GAN models. Although these experimentations are not directly discussed in the thesis, as they do not contribute to the immediate results, they played a crucial role in the selection process of the final base models. Additional factors such as time spent on implementation, refactoring time, and the quality of the code provided by the paper's authors were all considered in determining the most suitable models for the research.

Following subsections reveals the training configurations used in each experiment with the baseline and final models.

Baseline Models

When comparing different models, it is essential to maintain a consistent experimental setup across all models to ensure a fair comparison. However, specific models might have different optimal hyperparameters, which could lead to suboptimal performance under the same experimental setup. For instance, some models may perform better with a smaller batch size, while others might benefit from a larger one for optimal performance.

In this thesis, the four baseline models—CycleGAN, CUT, GcGAN, and StarGAN-v2— have been trained under a near-consistent setup of identical epochs, learning rate, batch size, train-test split, optimizers, and learning rate decay. In concrete terms, the default settings have been used, inspired by the original implemented training details used for each model in their respective paper, outlined in Section 7 [59], Appendix C [42], Section 4 [12], and Appendix B [8]. This means extensive exploration of architectural hyperparameters or model-specific data augmentations has not been pursued, as the primary goal of the early-stage experiments involves understanding what models are better suited for the image translation task under a unified setting.

The training configuration for each baseline model is triggered by the following command-line scripts in Appendix A and baseline configuration in Appendix C. In addition, the link to the original and thesis codebase for each model is attached below the script. The listed training configuration should allow the models to use the same parameters, though this is hard in practice as some rely on architectural model parameters.

Various pre-processing methods are added to each training session. The following list shows the common data-augmentation technique used for all the baseline models:

- **Resize(size=[256, 256], interpolation=bilinear, max_size=None)**
  Resizes the input images to a fixed size (256x256).
- **RandomHorizontalFlip(p=0.5)**
  Horizontally flips the input images with 50% probability.
- **ToTensor()**
  Converts the input images to PyTorch tensors.

- **Normalize(mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5])**
  Normalizes images by setting the channel-wise mean and the standard deviation.

The baseline models were trained on a single GPU rather than all three to ensure comparability (similar configurations as used in research papers), simplify the process, and maintain generalizability. Training on a single GPU allows for fair comparisons between models under the same hardware constraints and makes it easier to identify bottlenecks and limitations.

Each model was trained for 400 epochs using a learning rate $2e^{-4}$ with a linear decay starting at epoch 200 and a batch size of 8 to maximize VRAM usage. For optimizing the parameters of the neural networks, the models employed the Adam optimizer [27] with betas=[0.5, 0.999]. This optimizer is widely used and tested in various deep learning tasks and offers several benefits, such as fast convergence and robustness to noisy gradients.

Final Model

Oppose to the baseline training configuration, the training configuration for tuning the final model utilizing all three GPUs with the same baseline configuration C and only scaling the learning rate and batch size by a factor of three. This effectively means multiplying the learning rate, the number of workers[10], batch size, increasing from $2e^{-4}$ to $6e^{-4}$, 4 to 12, and 8 to 24, respectively. This ensures a similar training process while taking advantage of the additional computational resources the three GPUs provide. The tuning configuration aims to accelerate the training process and potentially achieve better performance with the larger batch size and increased learning rate.

The experimentation plan is structured in a way that first investigates the most common hyperparameters before gradually narrowing down to fine-tuning more specific parameters. The idea is to create a systematic and structured approach to hyperparameter tuning while following good practices in deep learning research.

The following explorations are to be explored and can be explained in the following way:

1. **Learning Rate**
   A reasonable learning rate greatly impacts the model's convergence and performance, making it crucial to explore various learning rates to find the optimal value.

2. **Image Resolution and Scaling**
   Different image resolutions for the dataset can impact the model's ability to capture relevant features and details. To continue with it confidently, an investigation to see if deviating from the standard 256x256 resolution is required.

[10]In PyTorch, the "number of workers" refers to the number of processes or threads used to load and preprocess data during the training process.

3. **Model Architecture and Initialization**
   Different architectures and weight initialization methods can influence the model's convergence and stability, making it essential to investigate their impact on performance.

4. **Data Augmentation and Preprocessing**
   Various augmentation and preprocessing techniques can enhance the model's robustness and generalization, making it valuable to experiment with different approaches.

5. **Reducing Color Channels**: Converting RGB images to grayscale can reduce the input complexity and improve model performance. It is an exciting experiment to explore how the impact of reduced color channels alters the translation quality.

6. **Manipulating the Loss Function**
   The idea of adjusting the final model's loss function to prioritize translation direction is found to be of interest, as the direction from the RGB domain to the IR domain is of particular interest, not the other way around.

4.4.3 Evaluation Details

Two widely-used evaluation metrics have been implemented to evaluate the performance of the various generative models in this thesis: FID and visual observations. As stated in Section 4.4.1, 711 images from each domain are used. Furthermore, during FID calculations, a batch size of 9 (a multiple of 711) was used to comply with the number of images available. The FID is calculated using the code implementation from the authors of StarGAN-v2 (see Appendix A).

One important consideration to keep in when using FID as an evaluation metric for this thesis is that the Inception network used to calculate the FID score is purely trained on ImageNet [9], which consists of colored images, even though this is not explicitly stated in their documentation. Therefore, while applying the FID metric to grayscale images, it is essential to note that this may not produce optimal results and will likely affect the final evaluation score.

As described in Section 3.4, visual observations involve qualitatively assessing the generated images by visually inspecting them to evaluate their quality, realism, and adherence to the target domain. This evaluation method allows researchers to identify artifacts, inconsistencies, or other issues that quantitative metrics like FID may not capture.

By combining the quantitative FID and the qualitative visual observations supported by an evaluation scheme, one can comprehensively analyze the performance of the unpaired image-to-image translation models in this thesis. This approach allows for a robust assessment of each model's strengths and weaknesses, facilitating informed decisions regarding the most suitable model.

5 Results and Discussion

This chapter presents both quantitative and qualitative results from various model experiments conducted. It provides a fair comparison of the tested models based on established metrics and benchmarks, discusses the strengths and limitations of each model, and offers insights into their performance in terms of image-to-image translation quality.

Some upcoming visualizations in this chapter are of high DPI, ensuring good image quality and detail. Therefore, readers are encouraged to zoom into the visualizations for a closer examination and better understand the presented data.

5.1 Model Capacities

The baseline training results, using the configuration in Section 4.4.2 and summarized in Table 5.1 below, provide an overview of the resources and performance associated with each model. Specifically, one can observe that the correlation between model size and training times is tightly coupled for each baseline model. A more detailed overview of the number of model parameters can be found in Appendix B.

Model	Training Time	Generator Size
CycleGAN	33h 04m	11.37M
CUT	26h 15m	11.38M
GcGAN	17h 56m	7.84M
StarGAN-v2	57h 29m	33.89M

Table 5.1: Comparison of time spent training and generator size

As one can observe from the various training sessions in Table 5.1, GcGAN has the shortest training time, while StarGAN-v2 requires the longest. It is important to note that a more extensive network size may contribute to longer training times and increased computational requirements.

5.2 Baseline Results

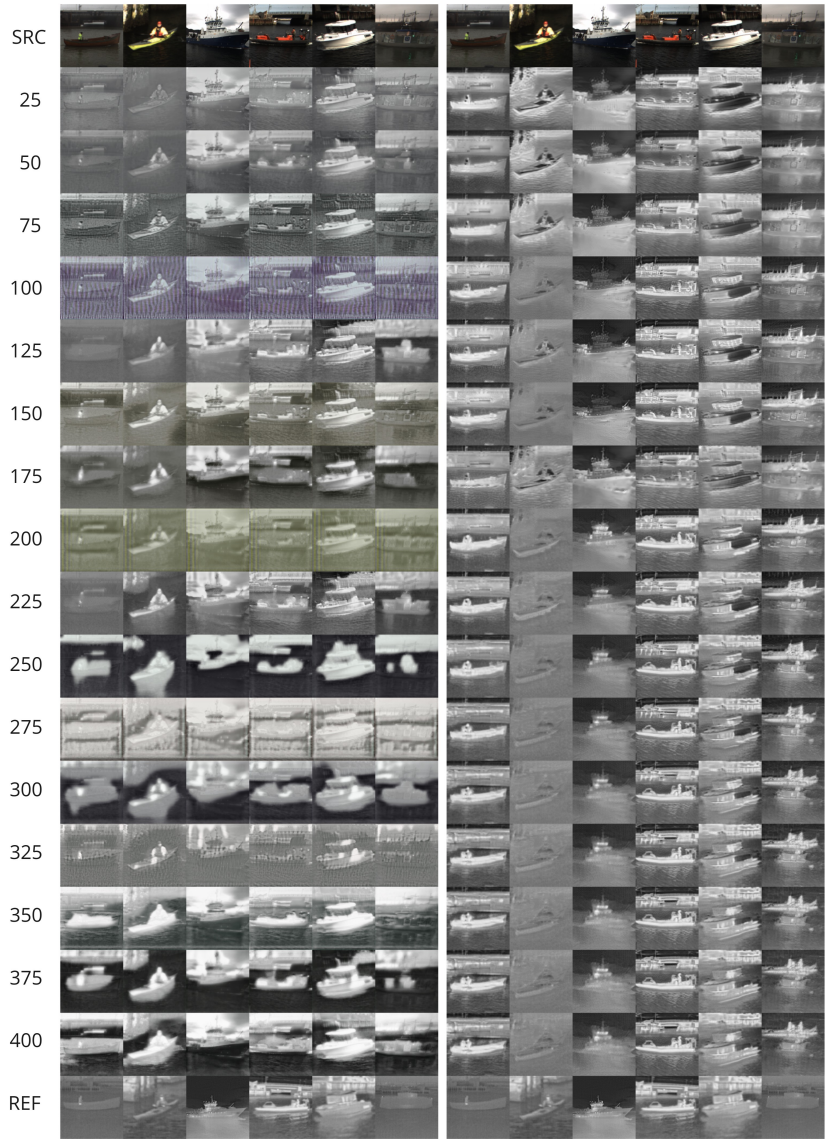


Figure 5.1: Baseline model comparison 1: CUT (left) and CycleGAN (right). The first row (source) presents six hand-picked RGB images from the validation set used to evaluate the translation quality for each object. The last row (reference) displays the closest real IR images, representing the desired appearance. The intermediate rows contain samples generated using various checkpoints from the training sessions. This setup remains identical throughout the chapter.

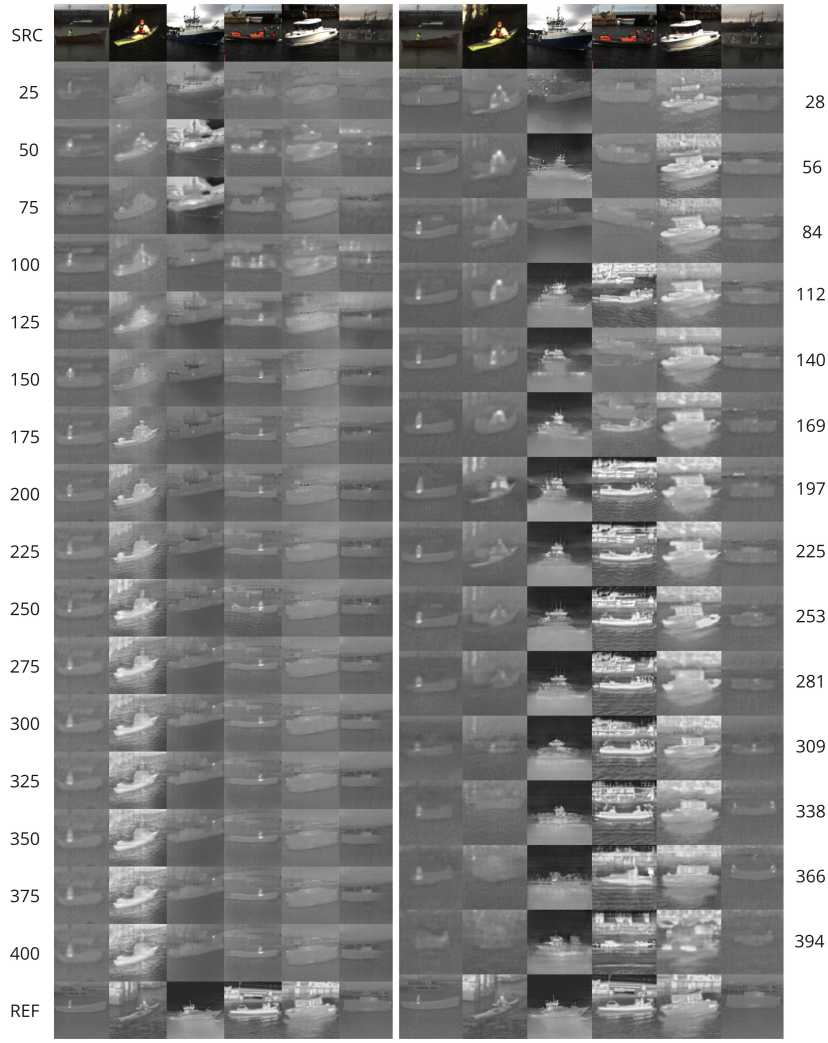


Figure 5.2: Baseline model comparison 2: GcGAN and StarGAN-v2. The row arrangement is the same as in Figure 5.1. It should be noted that StarGAN-v2 used iterations for sampling, leading to a discrepancy when converting iterations to epochs. However, this inconsistency likely does not significantly impact the final results.

The baseline model comparisons, displayed in Figures 5.1 and 5.2, show the various model improvements during the training session. By displaying epoch-by-epoch translation results, one can see how the models learn certain features over time. For evaluating the models with the evaluation scheme (from Section 3.4.2), the final epoch (400) is used as the primary observation indicator, with secondary observations being anything between 0 and 400. For readability purposes, the following sections use the notation of (1), (2), ..., (6) to the corresponding columns (1), (2), ..., (6) in the generated image-grid - e.g., wooden

boat (1), kayak (2), etc. By monitoring the generation result of each specific object, one can quite quickly notice deviations.

CUT

By observing the image progress in Figure 5.1 by CUT, it becomes apparent that pixel textures and the hidden outlines of the structure in the image are pretty preserved. However, the grayscale heat signature is not correct, resulting in overly bright color gradients, specifically for the unique objects centered in the image. Compared with the real image, objects such as the kayak (2) and white boat (5) are generated quite bleached, which does not sufficiently match the reference real infrared image. In terms of the structural positioning, most of the objects are well preserved. The objects in the image are translated without any significant change in position in the scene while maintaining proper shape and size. Observing the model, epoch by epoch, one could see that the model struggles with applying correct gradients and color changes, producing a mix of purple and yellow tints instead. The correct sharpness of the generated images is subpar, as too many details from the electro-optical domain are preserved. The blurred objects and edges around them, in conjunction with artifacts such as unnatural textures and color-bleeding surrounding them, resulting in a poor translation. In sum, by summarizing the results in Table 5.2, the model is given a score of 2.

Criteria	**Weight**	**Score**	**Weighted Scored**
Realism	0.25	2	0.50
Consistency	0.20	4	0.80
Detail Preservation	0.20	1	0.20
Sharpness	0.20	1	0.20
Artifacts	0.15	2	0.30
Sum			**2.00**

Table 5.2: Evaluation Scheme: CUT

CycleGAN

By viewing the image generations from CycleGAN, one can immediately see that some outputs, particularly the vessel (3) and speedboat (4), are, in fact, quite realistic. Using the speed boat as an example, one could observe that the image exhibits realistic colors, contrast, and brightness levels in significant similarity to the true infrared image. The background elements, such as the bridge and the water in the foreground, look moderately natural and plausible.

In terms of consistency, the physical positioning of the object is accurate, except for the research ferry (6). Purely observing the focused object's boundaries, they all (except 6) have the correct direction/pose considering the slight differences in the field of view and depth information between RGB and IR. As mentioned, regarding background elements,

the wave-formations in the water look realistic - though with subtle deviations. For instance, looking at the wooden boat (1), one could see that the water surfaces look plausible, perhaps with too much detail. However, as the dataset contains "two styles of infrared," one could accept either style compared to the real infrared.

This relates well to detail preservation, as surface details seem intact in some regions. Using the big vessel (3) as an example, one can see that the color gradient around the heated windows seems realistic. Another sign of this quality is the vessel's hull, generated as a solid color instead of a separation between white and blue.

In terms of sharpness, most of the object boundaries show the same level of sharpness/blurriness as the true infrared image. For instance, compared to the true equivalents - the speedboat (4) shows signs of a higher sharpness while the big vessel (3) is lower.

Prominent for CycleGAN is the presence of artifacts, specifically by comparing the wooden boat, kayak, and ferry. Multiple instances of the generated images exhibit uneven or unnatural textures. The addition of what looks like an extra person in the wooden boat, the kayak's lack of color, and the ferry's misplaced features all support this claim. In sum, by summarizing the results in Table 5.3, the model is given a score of 3.5.

Criteria	**Weight**	**Score**	**Weighted Scored**
Realism	0.25	4	1.00
Consistency	0.20	4	0.80
Detail Preservation	0.20	3	0.60
Sharpness	0.20	4	0.80
Artifacts	0.15	2	0.30
Sum			**3.50**

Table 5.3: Evaluation Scheme: CycleGAN

GcGAN

In evaluating the result for GcGAN, one can immediately see that most objects are translated into shape variations of the ferry (6), signifying an unrealistic translation. This might indicate that the model is suffering from mode collapse, a common issue in generative models as described in Section 2.2.3. In short, the model seems unresponsive to input variations. However, the color consistency appears to be accurate. While looking at the ferry (6), one can see similar levels of color and brightness, making it look natural to a certain degree.

Even though multiple outputs display the same object class (6), the object sizes and proportions seem consistent with the true input image. For instance, by looking at the speedboat (4), the object orientation, position, and size of the object in both the generated and the actual look relatively the same - which indicates some level of consistency.

Regarding detail preservation, the generated images do not retain essential features and information from the source images, as most images are incorrectly translated. With this in mind, the translation does still have color gradients intact.

For the correctly translated images, wooden boat (1), kayak (2), and ferry (6) all indicated a low level of sharpness compared to the real image. The image clarity is not maintained throughout the translation leading to an unclear and blurred expression.

Finally, the artifacts represented by the misplaced features (3-4-5) exhibit uneven or unnatural textures. In addition, even though the kayak (2) is correctly translated, the overall image is affected by color bleeding and bizarre color transitions leading, indicated by the higher brightness and the blurred edges around the central object. With all criterias evaluated, the final score of GcGAN is 1.85 as summarized in Table 5.4

Criteria	**Weight**	**Score**	**Weighted Scored**
Realism	0.25	3	0.75
Consistency	0.20	2	0.40
Detail Preservation	0.20	1	0.20
Sharpness	0.20	1	0.20
Artifacts	0.15	2	0.30
Sum			**1.85**

Table 5.4: Evaluation Scheme: GcGAN

StarGAN-v2

At various training epochs, StarGAN-v2 produces a few realistic translations, ferry (6) at epoch 140, wooden boat + person (1) at epoch 197, kayak (1) at epoch 225, and speedboat (4) at epoch 253. This indicates that the model can learn to a certain degree but struggles to be consistent for all classes simultaneously. However, there are often more failure cases than consistent translation leading to an overall poor realism impression.

Regarding consistency, the orientations and positions are also out of order for a few outputs. For instance, looking at the speedboat (4), the actual objects' position in the true input image is centered, while in the generated image - the boat is in the upper section of the image in the early training stages. With this in mind, it is still accurate for other objects such as the wooden boat (1), kayak (2), and white boat (5).

By observation, the translated images have their color gradients intact to some degree. Nonetheless, the generated features are still incorrect, with low to no information preserved from the input images.

As for sharpness, the generated images exhibit a slightly lower level of sharpness compared to the true infrared. Though the object is correctly translated (1), the edges in the generated image are not equally crisp and well-defined resulting in blurring during translation.

Unnatural textures, such as the outline of two persons on the ferry (6), contribute to the ghosting effect of the translation. There are also a lot of inconsistent features from one epoch to another. For instance, the details tend to disappear at certain stages, like one can observe early for the vessel (3) or late for the kayak (2). In short, in addition to

the previous deficiency mentioned above, these misplaced features result in a score of 1.65, as summarized in Table 5.5.

Criteria	Weight	Score	Weighted Scored
Realism	0.25	2	0.50
Consistency	0.20	1	0.20
Detail Preservation	0.20	2	0.20
Sharpness	0.20	2	0.40
Artifacts	0.15	1	0.15
Sum			**1.65**

Table 5.5: Evaluation Scheme: StarGAN-v2

5.2.1 Summary

In addition to the evaluation schemes above, the baseline model comparisons, displayed in Figure 5.1 and 5.2, one can see how the FID scores fluctuate as each checkpoint is evaluated on the 711 validation images (see evaluation details in Section 4.4.3) in Figure 5.3

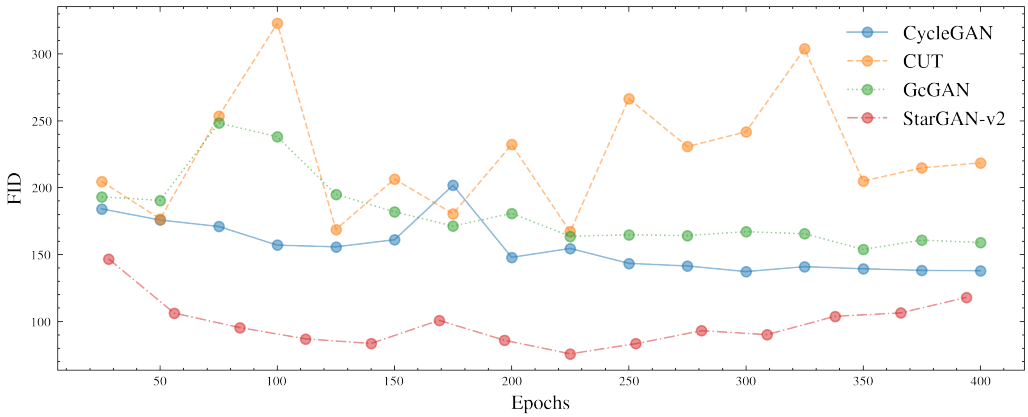


Figure 5.3: Showing FID progression during training

Using the last epoch for evaluation, one can visualize a correlation in performance by combining the results from the visual evaluation schemes with the final FID scores. An illustrative scatter plot, shown in Figure 5.20, exhibits the relationship between the two evaluation metrics for each of the four models under experimentation: CycleGAN, CUT, GcGAN, and StarGAN-v2.

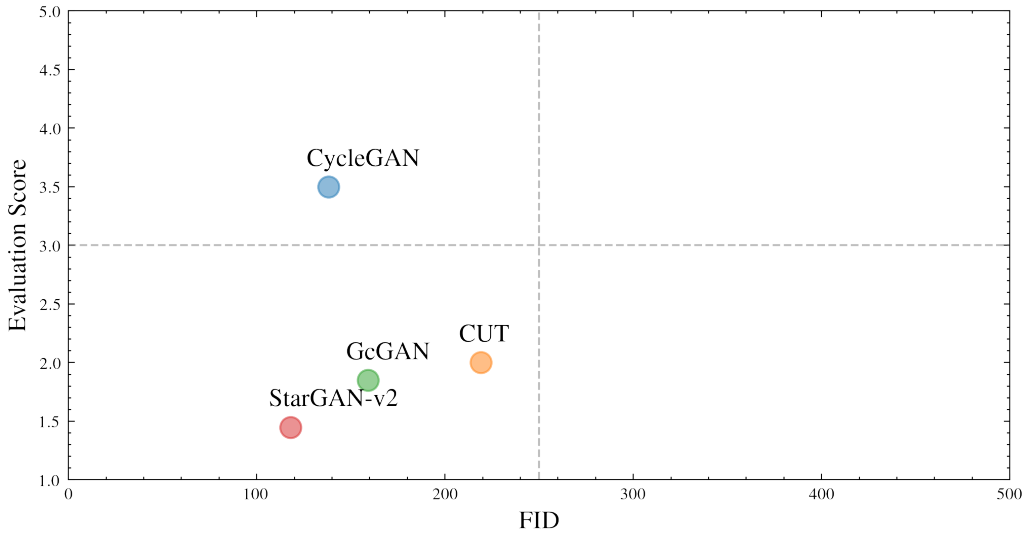


Figure 5.4: Visualizing baseline model performance by comparing evaluation score with FID. The top-left section indicates higher performance, lower-right section indicates lower performance. Note the limit for FID is set to an arbitrarily large limit, while the evaluation score follows the Likert scale.

Upon close inspection of the plot, a clear negative correlation between FID scores and visual observation scores is noticeable, indicating that a decrease in the FID score corresponds to an improvement in the visual quality of the generated images. This correlation aligns with theoretical expectations since lower FID values often reflect a superior performance in terms of visual quality.

In short, by analyzing the models, it becomes evident that StarGAN-v2 outperforms the other models in the present baseline experimentation, with the lowest FID score of 118. For comparison, CycleGAN got a score of 143 and also had the highest visual observation score of 3.50. Since the FID score and visual generations do not correlate sufficiently, the best performance is demonstrated by CycleGAN, as visual observation is the most important indicator. This shows that the CycleGAN inhabits the more significant potential of success, given the baseline training settings as previously described in Section 4.4.2.

5.3 Tuning Results

Based on the performance analysis from the experiments conducted in Section 5.2, CycleGAN emerged as the preferred model for continued tuning in this research. It exhibited the highest visual observation score while achieving a competitive FID score and utilized adequate resources, notably in training time.

The transition from a single GPU to a setup employing three GPUs was a significant

development in the model training process. This change significantly reduced training time by approximately 47%, decreasing from 33h 04m (refer to Section 5.1) to 13h 47m. This modification illustrates the benefits of parallelizing the training process across multiple GPUs, facilitating quicker convergence and more efficient hyperparameter exploration.

With this improved efficiency, the final model of CycleGAN was further tuned to enhance its translation capabilities. As described in the experimentation plan, Section 4.4.2, the following subsections will alleviate and highlight the main results from the various tuning studies. For clarity and brevity, only the final and best checkpoint (epoch 400) for each experiment is used in image generation comparisons, as seen below. Metrics and image generations were logged every 25th epoch during each experiment run, providing comprehensive data for analysis. More information about the frequency of logging can be found in the training configuration in Appendix C.

5.3.1 Learning Rate Exploration

Figure 5.5 below visualizes the relationship between various learning rates and their respective FID scores. The results indicate that lower learning rates contribute to less spiky and more consistent evaluation scores. Slower learning rates often lead to longer training sessions. For this particular case, the difference between the lowest and highest learning rate is ~ 15 minutes.

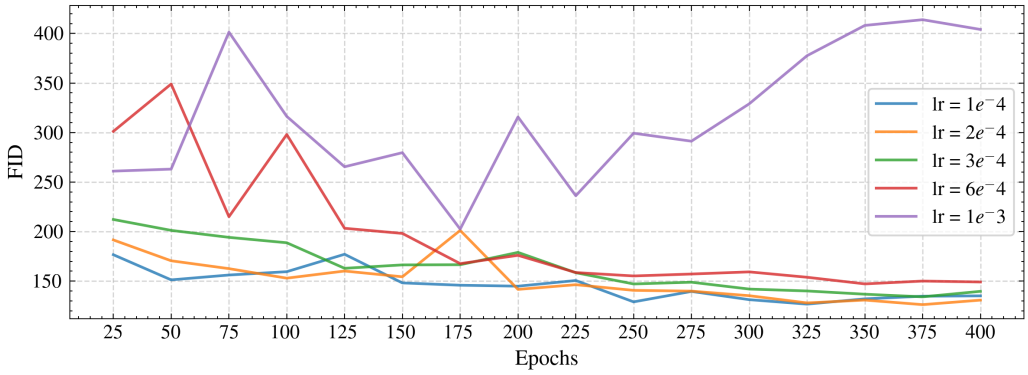


Figure 5.5: LR Experimentation: Illustration compares FID score and varying learning rates. The vertical dashed line at epoch 200 indicates the starting point of the linear learning rate scheduler. Note: two identical configurations can produce two slightly FID plots. Sources of randomness include data shuffling, weight initialization, and batch sampling. When using GPUs, setting the random seed might not guarantee the same results due to non-deterministic implementations of some GPU operations.

Figures 5.6 and 5.7 illustrate the diminishing returns of training the model for more extended periods with the same learning rate. From 400 to 600 epochs, the model's

visual quality visibly deteriorates, suggesting an overfitting scenario where the model no longer generalizes well to unseen data.

Figure 5.6: Model generations: baseline model trained with learning rate = $1e^{-4}$ for 400 epochs. The row arrangement is similar to the previous figures.

This reinstates that a sufficient length of 400 epochs is a practical choice for this model and dataset, balancing training time and the output quality. It becomes evident that extending the training duration beyond this point does not bring additional benefits but rather contributes to a degradation in model performance.

To investigate this further, a model with similar configurations was trained for a more extended period - totaling 1200 epochs. Interestingly enough, the FID score continued to improve (decline). However, observing ongoing image generations during training showcased profound mode-collapse, where one object often translated into another. As one can observe, by looking at the three samples (A), (B), and (C) in Figure 5.8, the model confuses objects and generates incorrect representation. Looking at case (A) with a kayak as input, the model generates a mix of a kayak and the same infrared style as commonly seen for the wooden boat. For case (B), the model generates something more similar to the white boat when the input is the research vessel, while for case (C), the speedboat is translated into a representation much like a wooden boat instead.

While a learning rate of $1e^{-4}$ was optimal in this context, it may not be the most suitable for all models and datasets. The optimal learning rate varies depending on the dataset's complexity and characteristics and the selected model's architecture. Training for prolonged periods did not result in any immediate improvements, which might indicate that the model has reached its capacity to learn or effectively generalize from this specific dataset. In the particular scenarios above, even with a lower learning rate and increased training duration, the model seemed to have hit a performance plateau, and further training even led to degradation in the form of mode collapse. Measures such

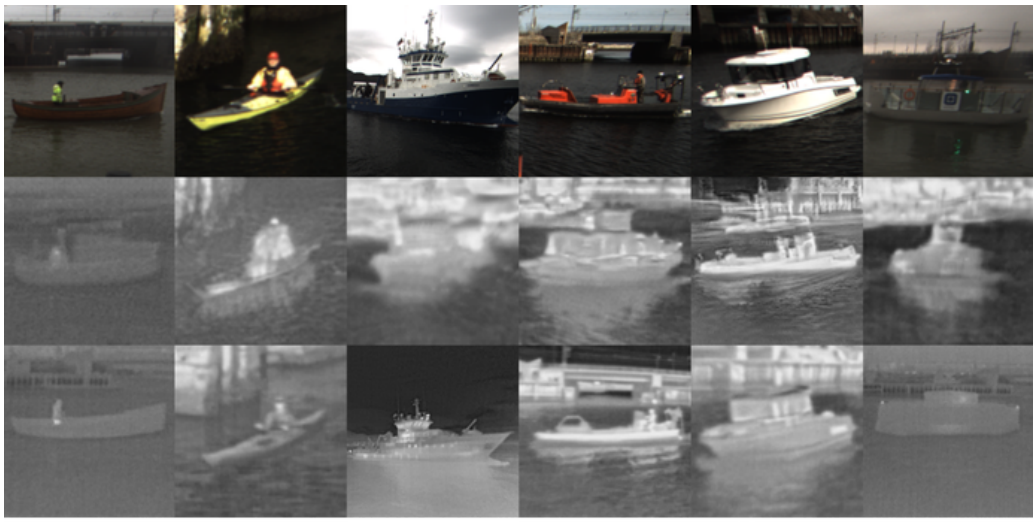


Figure 5.7: Model generations: baseline model with trained with learning rate = $1e^{-4}$ for 600 epochs.

(A) (B) (C)

Figure 5.8: Three randomly sampled generations outputs during the last training phase for 1200 epochs. The model used the default configuration. Case (A) is given a kayak as input, case (b) is given a vessel as input and case (C) is given the speed boat as input.

as early stopping are often implemented for deep learning models, typically monitoring a particular metric. As neither the model's loss metrics nor FID score does not correlate sufficiently with image quality, this becomes a difficult challenge to solve. Moving forward, future experiments will utilize a learning rate of $1e^{-4}$, substituting the original rate of $6e^{-4}$. This decision is based on the experimental results indicating that this rate provides optimal performance in terms of FID scores and the visual quality of the generated images. For the sake of brevity, subsequent references to the 'baseline' in this context will refer to the model configuration with this adjusted learning rate.

5.3.2 Exploring Image Resolution

Using high-resolution images allows for potentially more detailed and precise translations but also demands substantial computational resources. Previous efforts in the Specialization Project used full-size images that often contained a visual context without

boats, kayaks, vessels, etc. As this thesis works with image crops of objects, investigating appropriate crop resolutions becomes an interesting topic, as the data exploration revealed various sizes in both the electro-optical and infrared domains.

Figure 5.9 visually represents this exploration, highlighting the varying FIDs scores resulting from the training. These scores strongly correlate with the image generations presented in Figure 5.10, further emphasizing the potential impact of image resolution on model performance.

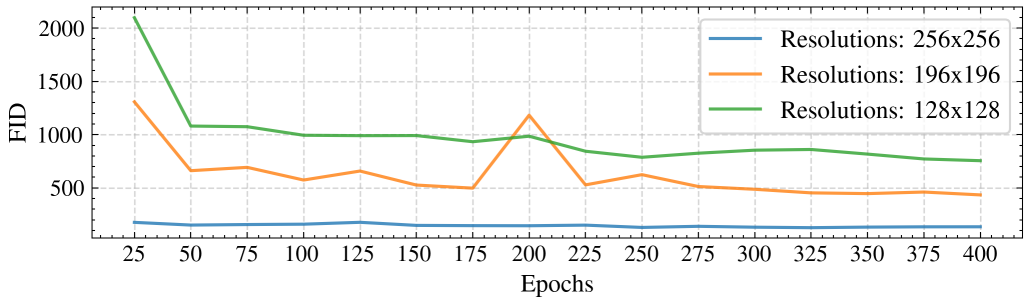


Figure 5.9: Image generation with three different models, where each model uses image at various resized resolutions from the dataset.

Interestingly, modifying the resolution of the images had a direct impact on the models' performance. Compared to the original size of 256 px, the 196 px and 128 px do not yield additional performance. The current generator architecture in the CycleGAN model, a ResNet[14]-based generator with 9 ResNet-blocks, is explicitly designed for 256x256 images. Hence, comparisons with lower resolution are not entirely fair, as one could benefit from using fewer ResNet blocks when using lower-resolution images.

The thesis's initial decision to use a resolution of 256x256 was driven by the desire to maintain compatibility with industry standards. However, this resolution may not be optimal for the specific characteristics of the dataset. For instance, working with an average resolution of 200 px for RGB images and 100 px for IR images, as seen in Tables 4.2 and 4.3, suggests that we could potentially benefit from conducting additional experiments to strike a balance between the different domains. The choice to retain the default image resolution of 256x256 will be adhered to for the baseline model in subsequent experiments.

5.3.3 Exploring Model Architecture and Initialization

The upcoming section tests various model architectures and weight initialization strategies to understand their influence on the generative model's performance. Specifically, the experiments investigate different generator architectures, such as ResNet and Unet (briefly described in Section 2.2.2, and alternative activation functions and weight initialization methods.

Figure 5.10: The first (source) and last (reference) row presents six hand-picked RGB images from the validation set used to evaluate the translation quality for each object in their respective domain. The intermediate rows show baseline model generations trained on (1) 256x256 images, (2) 196x196 images, and (3) 128x128 images sampled from the final checkpoints.

Figure 5.11 illustrates the result of these experiments. While the models seem to converge to similar scores, there are slight differences. Notably, the UnetGenerator, denoted in orange, exhibits minor fluctuations in its early checkpoints.

From the generations, studying Figure 5.12, it seems evident that using the profound ResNet-architecture in the generator appears to contribute to the best visual performance, comparing (1), (2), (3), and (4). Comparing (1) ResNetGenerator and (2) UnetGenerator, one can observe the following indications. By assessing class by class, the overall impression of the ResNetGenerator seems to reign in performance. For instance, by observing column 1, the generation of the wooden boat for each architecture differs a lot. The ResNetGenerator creates something similar to the reference infrared in realism and overall positional structure, contributing to a decent translation. Compared with the UnetGenerator, the same class deviates from its original features and converges to features

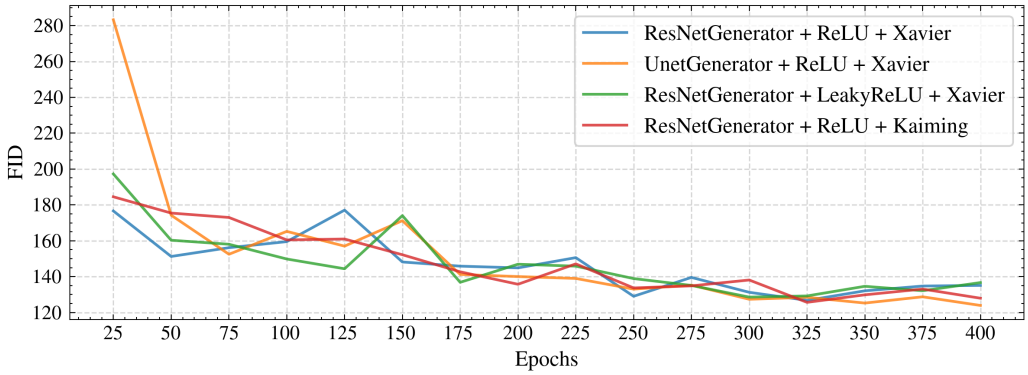


Figure 5.11: FID scores with varying generator architecture and weight-initialization method. Blue indicates the baseline model, which uses the generator's ResNet architecture (default). The orange represents the Unet-architecture in the generator instead. The green represents the baseline model but with LeakyReLU instead of ReLU (default). The red represents the baseline model with Xaiming weight-initialization instead of Xavier (default).

more similar to the ferry than the wooden boat itself. Only the positional structure of the wooden boat seems to be in place.

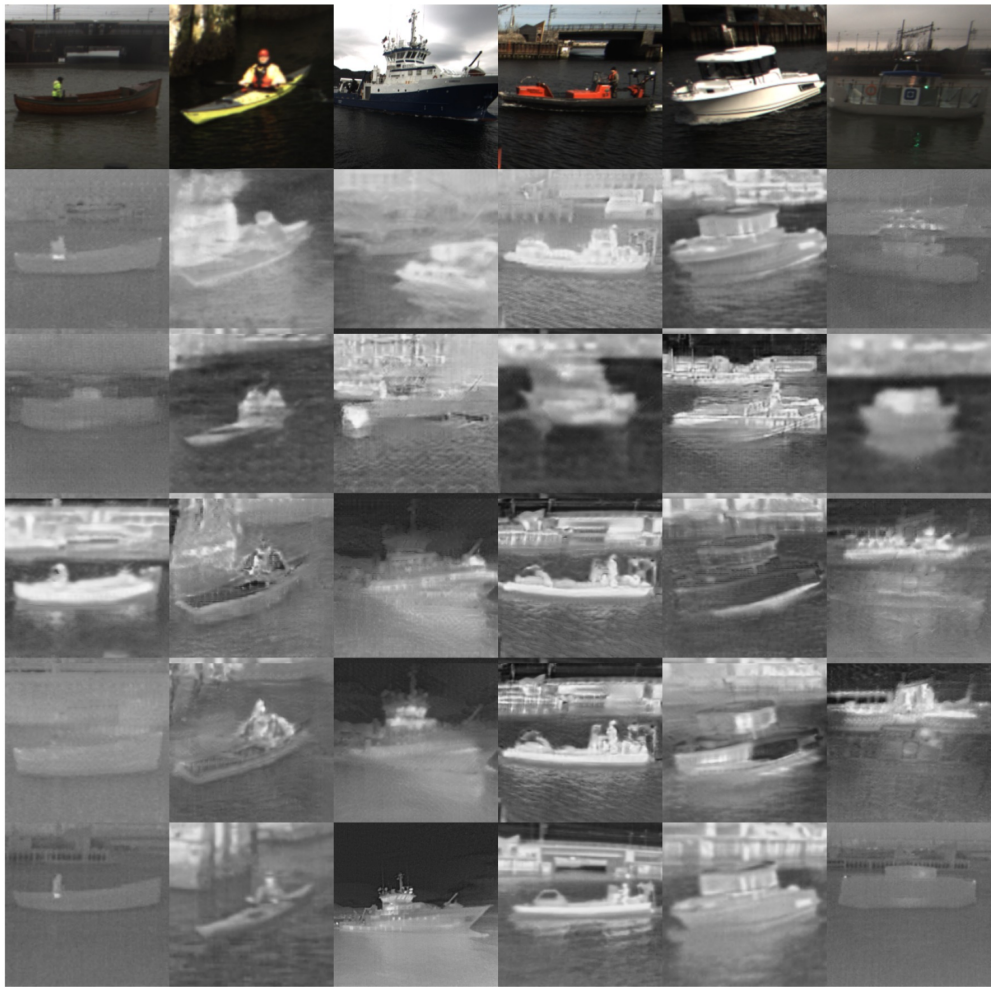


Figure 5.12: Image generations with varying generator architecture and weight initialization method. Like previous image class generations, the top row consists of the source RGB image, while the last row is the reference infrared image. Rows in between are the best and final checkpoint (epoch 400) from each experiment;

(1): ResNetGenerator with ReLU and Xavier (Baseline)
(2): UnetGenerator with ReLU and Xavier
(3): ResNetGenerator with LeakyReLU and Xavier
(4): ResNetGenerator with ReLU and Kaiming

Continuing with the ResNetGenerator for both weight initialization experiments, the baseline configuration continues to reign on top, similar to previous results. When using the ResNet architecture in the generator, its common practice to use ReLU as the nonlinear activation function. Since ReLU activations are predominantly used, using Kaiming (He) [15] initialization is recommended for weight initialization by default. However, it is

essential to note that in the conducted experiments, the Kaiming initialization with ReLU activations is similar to the default Xavier initialization regarding visual performance. It generates realistic vessels and speed boats but fails to translate the wooden boat and ferry, as the necessary heat signatures of the person are gone, and the logo on the ferry is still intact.

From the analysis presented above, several key findings emerge: the ResNet generator consistently outperforms the Unet-generator, and the default settings employing ReLU as the activation function remain superior. Both weight initialization methods demonstrate similar performance, producing satisfactory visual results. These findings underscore that, at least for this specific dataset and task, the default settings are typically the most effective choice. Additionally, the results show that even when the default Xavier initialization is selected, the Kaiming initialization coupled with ReLU activations performs comparably well. However, it's crucial to emphasize that these results are specific to the current dataset and might lack generalizability to other datasets or tasks. Further experimentation is necessary to pinpoint the optimal architecture and weight initialization method for varying scenarios. According to the authors of the CycleGAN paper, they have also observed the best results using the ResNet architecture.

5.3.4 Data Augmentation techniques

The scale varies considerably due to the significant discrepancy in image resolution across domains within the dataset. As discussed in Section 3.2.1, the scale difference approximates around 100 pixels. When resizing all images to a uniform resolution, 256x256 pixels, in this case, each domain encapsulates two distinct sets of resolution - one higher and one lower.

Resizing images to a uniform resolution inevitably affects image quality. For higher-resolution images, downsizing to 256x256 pixels may lead to a loss of detail and potentially introduce artifacts. Conversely, resizing to 256x256 pixels for lower-resolution images may cause pixelation and a general loss of clarity due to upscaling.

This experiment adds Gaussian noise to the RGB images to emulate the degraded quality often seen in infrared images. This was an attempt to bridge the gap between the two different resolution types in the dataset. The impact of this data augmentation technique on the FID scores is shown in Figure 5.13.

As one can see in the figure, the RGB added noise is controlled by parameters used in the first noise experiment with mean $\mu_1 = 0$ and standard deviations $\sigma_1 = 1$ and in the second with mean $\mu_2 = 0$ and standard deviations $\sigma_2 = 10$. Immediate observation suggests that increasing the standard deviation of the noise makes it more challenging to learn while using a more conservative value of 1 seems to be competitive with the baseline model. To test whether this correlates to image quality, one can compare with the generations in Figure 5.14.

Comparing the baseline (1 - no noise) with the added noise models (2) and (3), one can see that; generally, the baseline model generates fewer failure cases. Comparing

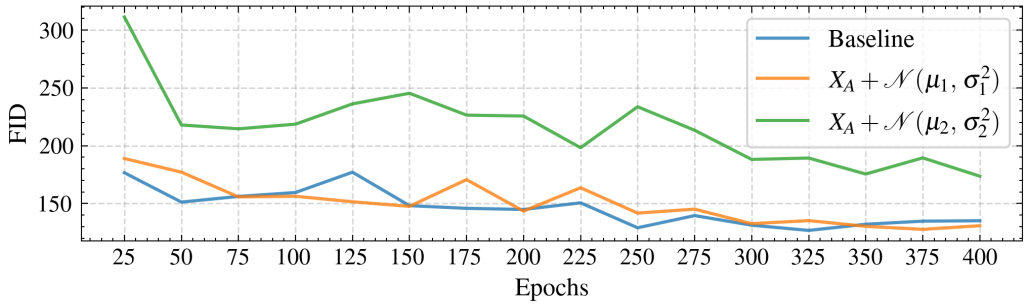


Figure 5.13: Effect of adding data-augmentation to RGB images. The comparison shows the baseline (blue) with no noise, baseline with noise $\mu_1 = 0$ and $\sigma_1 = 1$ (orange), and baseline with noise $\mu_2 = 0$ and $\sigma_2 = 10$ (green).

Figure 5.14: Image comparison of baseline model generations with varying RGB noise. Specifically, the comparison shows (1) no noise, (2) added noise with $\mu_1 = 0$, and $\sigma_1 = 1$, and (3) added noise with $\mu_2 = 0$ and $\sigma_2 = 10$.

the vessels (column 3), one can see that model (3) captures more realistic features than (1) and (2) but does not preserve the heat signature around the bridge, resulting in an overall inconsistent image translation. The model (3) also outputs the most realistic speed boats (column 4), where the shape, size, and relative position seem correctly placed with relatively correct heat signatures.

5.3.5 Exploring reduced Color Channels

To investigate the impact of color information in the electro-optical images, this subsection explores the idea of reducing from 3-channel (RGB) to 1-channel (grayscale) images. By converting the images to grayscale, the model relies solely on the intensity values rather than color in both the source (input) and target (output) domains. Observing the FID score results in Figure 5.15 between the baseline and the grayscale model, they both output similar results.

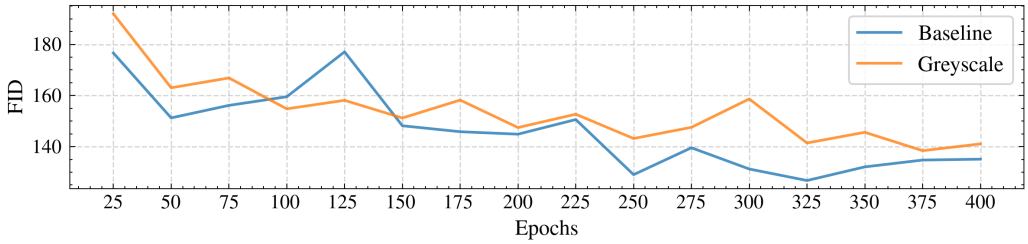


Figure 5.15: Effect of using purely grayscale images. Specifically, the input channels are changed from 3 to 1.

Comparing the baseline (1) and grayscale (2) models, column by column, both models generate decent results with a few failure cases. As ferries are the most frequent class in the dataset, it is natural to expect such generations to be superior. Comparing (1) and (2), the baseline model generates more or less something akin to the ferry, with a minor loss of preservation at the ferry's front. While for the grayscale model, it struggles with capturing the general features of the ferry, resulting in an obscure output with deficiencies such as incorrect heat signatures and infrared textures.

Figure 5.16: Visual generations of the various class objects in the dataset, Specifically, row 2 and 3 represents (1) the baseline model trained on 3-channel RGB input images, and (2) a grayscale model, trained on 1-channel black and white input images.

5.3.6 Regularizing the Loss Objective

The following section investigates whether favoring a translation direction is beneficial. Specifically, prioritizing translation direction in the cycle loss for the CycleGAN model. In this case, the direction of the translation in the CycleGAN's cycle loss is prioritized from the RGB domain to the IR domain. Figure 5.17 shows the respective scores, where regularizing the cycle loss results in worse initial scores while still converging to approximate the same score, as the baseline, at the end of the training regardless.

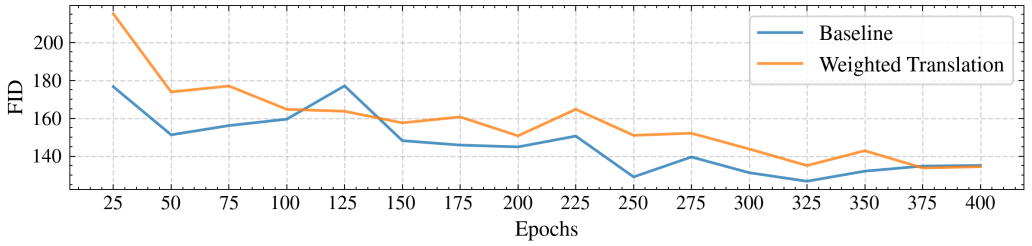


Figure 5.17: Altering the weights of the cycle consistency loss (3.2) with $\lambda_A = 15$ for the forward cycle loss and $\lambda_B = 5$ for the backward cycle loss.

One reason for changing the weight values in the cycle consistency loss was to prioritize the translation from the RGB domain to the IR domain by assigning a higher weight to the forward cycle loss (λ_A) compared to the backward cycle loss (λ_B). The primary focus of the study is to achieve high-quality translations from RGB to the IR domain, and the weight adjustments reflect this objective. Giving a higher weight means that the model is penalized more for errors in translating from the RGB domain to the IR domain than vice versa. The higher weight on λ_A encourages the model to pay more attention to producing accurate translations in the forward direction (RGB to Infrared).

Interestingly enough, regularizing the cycle loss generates similar images as the grayscale model in Figure 5.18. Notably, one can observe the same color intensity for the wooden boat (column 1), slightly better textures for the vessel (column 3), and close to identical representations of speed boats (column 4). This could be hints of stagnation, as the multiple models seem to converge to similar features - indicating a lack of diversity in the generated images, a potential sign of overfitting, or an indication that the models are reaching a limit in their capacity to learn from the given dataset.

Figure 5.18: Image generations: comparing two models, (1) the baseline model and (2) the cycle-loss regularized model, by visualizing six unique object generations.

Furthermore, one can observe by comparing the baseline (1) model with the regularized model (2) that some classes are generated better than others. For example, (1) produces more realistic wooden boats and ferries, while (2) produces more realistic ships and speed boats. One might instinctively assume that the quality of generated images for

each class would be proportional to the number of training images for that class in the dataset. This is, of course, a simplification, as the complexity of the various scenes varies greatly. For instance, the kayak is one of the least represented classes but arguably is in the most contrasted scene with little-to-none other objects surrounding it. Comparably, the ferry, one of the more frequent objects, is, by observation, a more complex object with more unnatural features, such as illuminating lights and see-through walls. Hence, due to its complexity combined with various other background elements such as poles, high-voltage lines, and bridges, it naturally becomes harder to translate.

5.3.7 Final Model

Based on the various experiments and analyses presented previously in the results, the most effective model for image-to-image translation using the given dataset is a CycleGAN model with a ResNet-based generator and the default configuration settings. This includes a ReLU activation function, Xavier weight initialization, and an optimal learning rate of $1e^{-4}$. With these settings, the model produces the best quality images and competitive FID scores while achieving a balance between training time and translation quality.

Using the selected final model, the following image generations can be visualized. Moreover, as one can observe in Figure 5.19, there is a gradual improvement in the quality of image translations as the model progresses through the training epochs.

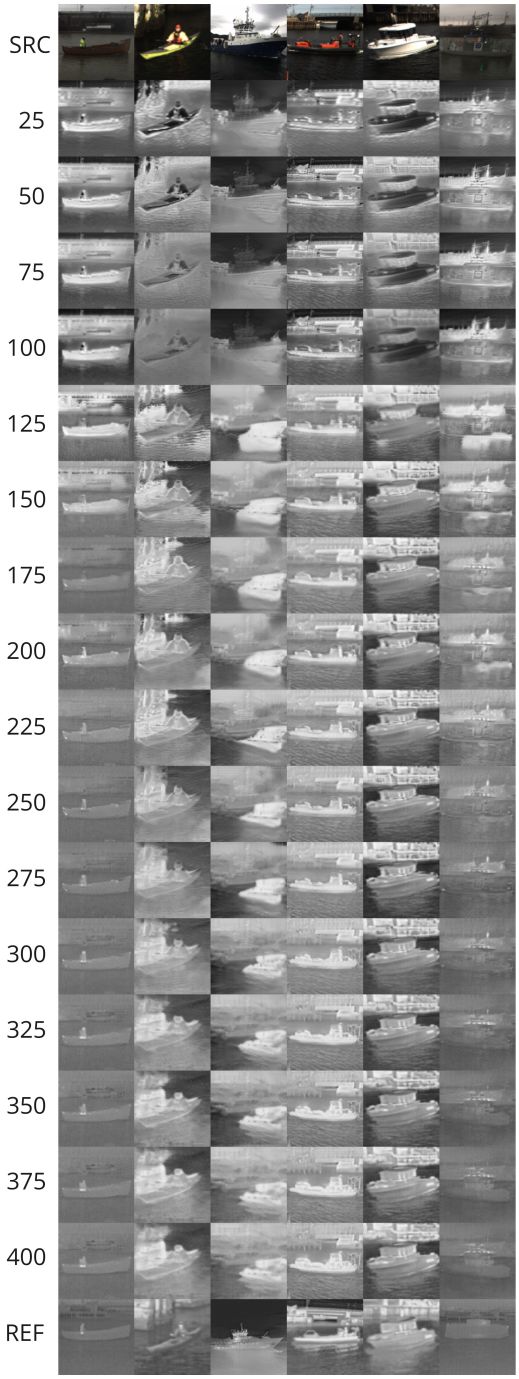


Figure 5.19: Illustration shows progressive image generations from the final model during training. The first (SRC) and last (REF) rows present six hand-picked RGB images from the validation set used to evaluate the translation quality for each object in their respective domain. The intermediate rows show sampled model outputs.

To expand on the evaluation of the final model, the visual scheme used earlier is reused, as described in Section 3.4.2. To recall, the model is evaluated on the following criteria: realism, consistency, detail preservation, sharpness, and artifacts. As previously stated, this is a subjective process; thoroughly scrutinizing every aspect of the images is quite challenging. Hence, a systematic review process is followed using an evaluation scheme. The final epoch (400) is used as the primary observations, with secondary observations being anything between 0 and 400. A time-lapse video of all sampled images during training can be found in Appendix E.

Observing the output progress makes it apparent that some generations look plausible. At first glance, the wooden boat (1), the speedboat (4), the white boat (5), and the ferry (6) look like something that could represent an infrared object. A more detailed look indicates that the wooden and white boats might look the most realistic out of the four, with similar heat signatures as the reference image. Poor heat signatures can be seen for the kayak (2) and the vessel (3), as the overall translation could be more precise towards the later stages of training. Comparing initial samples with later samples, the model learned to invert colors, then transformed them into more plausible grayscale-looking images. For instance, at epoch 25, the white boat is translated to black - then progressively altered into nuanced grayscale, similar to the true image.

The physical positioning of the objects is preserved, as the outline of each object matches the input images respectively. The field of view and distance seems also preserved, as one can notice with the kayak (2). Comparing the source image with the reference image, the kayak looks more zoomed in. It contains less background information than the reference, which is also the case for the generated image. This becomes evident as one can see the size difference between the row at epoch 400 and the reference row.

Immediately, by observing the wooden boat (4), one can see that the surface details in the image are pretty aligned with the reference image. Specifically, the infrared style is well-captured, leading to a believable translation and overall impression. Looking at the ferry (6), one can notice that the model struggles a lot in the early stages since it fluctuates between various heat signatures and details but learns the important ones over time, resulting in a more or less decent infrared ferry, when not considering the outline contour of the ferry.

Regarding sharpness, almost all object edges and textures seem plausible and radiate the same level of blurriness/sharpness as the true infrared. Ignoring the failed heat signatures for the kayak (2) and the vessel (3), the rest looks close to the reference.

Slight inconsistent textures or ghosting effects for the kayak (2) and the vessel (3) result in an unnatural translation. Looking at previous epochs, one can notice that from epochs 100 to 125, a sudden irregular heat signature is generated and causes anomalies in later generations to appear. For the other classes, the widespread presence of artifacts is not significant.

In short, by summarizing the evaluation of the visual performance in Table 5.6, the

model is a given a score of 3.75.

Criteria	Weight	Score	Weighted Scored
Realism	0.25	3	1.00
Consistency	0.20	5	0.80
Detail Preservation	0.20	3	0.60
Sharpness	0.20	4	0.80
Artifacts	0.15	4	0.30
Sum			**3.75**

Table 5.6: Final Evaluation Scheme: CycleGAN

Onwards, one can observe minor improvements overall by comparing tuning results with baseline results. By looking at Figure 5.20, one can see that both CycleGAN (blue) and CycleGAN* (purple) perform well, with CycleGAN* taking the overall lead in terms of FID score and visual evaluated score, at 135 and 3.75 respectively.

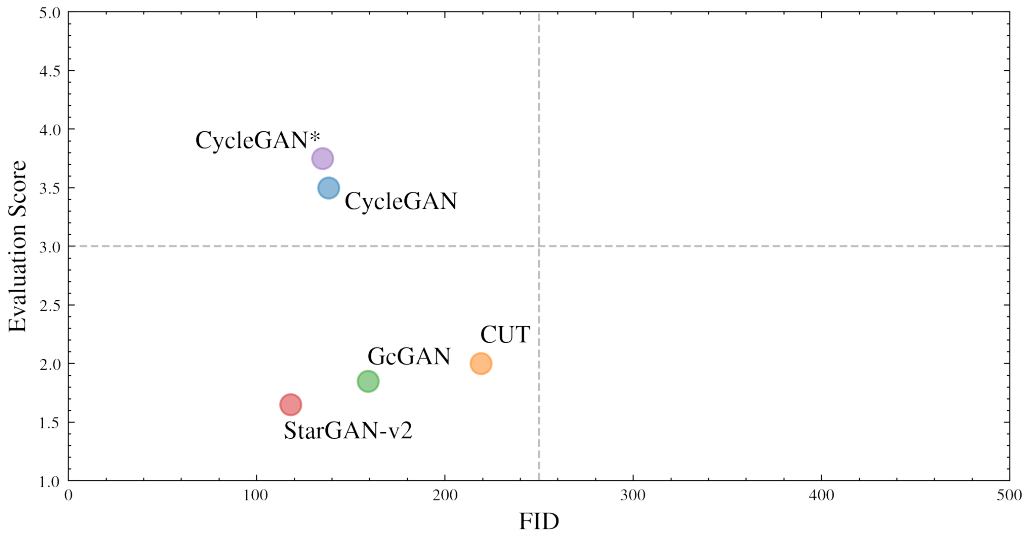


Figure 5.20: Visualizing final vs. baseline models by visual evaluation score and FID. The top-left section indicates higher performance, lower-right section indicates lower performance. Note the limit for FID is set to an arbitrarily large limit, while the evaluation score follows the Likert scale.

5.4 Discussion

From the research conducted in this study, key findings emerge.

The CycleGAN model and the various explorations from the parameter tuning formed a significant part of this study. The experiments aimed to tune the CycleGAN model yielded discoveries on multiple fronts. First, it was shown that adjusting learning rates impacted image translation performance, with a learning rate of $1e^{-4}$ seen to be the most effective choice. Second, the findings on various model architectures at least indicate that the ResNet generator outperforms the Unet generator. Continuing with default settings, such as ReLU activation and Xavier weight initialization, generally provided the best results.

A further interesting finding is that changing the image resolution did not necessarily improve performance, leading to a decision to maintain the original 256x256 resolution. Additionally, the results demonstrated that regularizing the forward and backward cycle loss in CycleGAN's objective did not yield significant improvements in image-to-image translation quality as initially hypothesized. Nor did the measure of incorporating data augmentation techniques and different weight initialization methods.

The results indicate that using a modified CycleGAN model with a ResNet-based generator, ReLU activation, Xavier weight initialization, and a learning rate of $1e^{-4}$ balances training time and performance. These modifications deliver better image quality in most objects studied within the presented dataset, achieving a final visual evaluation score of 3.75. However, it is essential to recognize the training limitations of the work presented, such as the lack of concrete methods, such as early stopping to guide the training process, which may be beneficial to avoid overfitting or model degradation.

In comparison to prior studies, such as the Specialization Report [38] reporting an evaluation FID score of 195 using the same model architecture, the decision to use smaller image crops as opposed to full-resolution images significantly improved the overall visual quality and led to a final FID score of 135. However, the dataset itself might still present learning limitations due to the misalignment and disparity in representations between the electro-optical and infrared camera outputs, which reflect different perspectives of the same underlying scene. These differences between the two domains are critical issues preventing the accomplishment of better scores. Furthermore, the dataset lacks sufficient diversity (e.g., objects with various backgrounds) and representatives (e.g., multiple wooden boats), leading to overfitting and generalization issues.

Accounting for the dataset's underlying challenges, the minor improvements achieved by the final model are valuable. This demonstrates that additional tuning and experimentation allow the model to learn meaningful features and patterns in the dataset, despite the constraints. However, one should note that the improvements are limited, and the model might have reached its maximum potential, given the data available.

The complexity and variability of the image data, coupled with the limited dataset size of 2840 images per domain, may impose constraints on the model's learning capabilities. Despite careful parameter tuning, the model could face difficulties detecting subtle patterns, differentiating between subtle differences, and generalizing to unseen data. These limitations, made worse by potential imbalances in the dataset, could hinder significant improvements in the model's performance.

The method of assessing performance via evaluation schemes and metrics might only be partially appropriate as they could fail to encapsulate the many subtleties tied to the visual quality and various aspects of the translated images. This might lead to less than optimal tuning of the model's parameters. While our previous decision to utilize the FID metric enables comparison with standards set by the community and the industry, there may be more suitable choices for this dataset. Listed as a disadvantage, in Section 2.2.3, the metric has a bias since it requires typically big sample sizes to estimate the value reasonably. The pretrained InceptionV3 model, which uses color images (RGB) from ImageNet and FID relies on, is used during the validation stages on two sets of grayscale images - the real and generated. In conjunction with the sample size, this aspect is believed to introduce certain biases or limitations in the evaluation process.

Various tuning experiments of the model were tested, yet improvements over the default configuration remained minimal. This could be attributed to the already well-optimized CycleGANs model's default parameters, as the paper authors have extensively researched and optimized for a wide range of datasets. Therefore, the default settings may already be close to the optimal values for this dataset.

6 Conclusion

This thesis investigated the potential of unpaired image-to-image translation methods applied to co-localized electro-optical (RGB) and infrared (IR) image domains in the maritime scenery. The research objectives were accomplished through exploratory data analysis and dataset creation, various model implementations, and evaluation and optimization of the top-performing models. Existing unpaired image translation methods and techniques were thoroughly reviewed, leading to the selection of CycleGAN as the most suitable model for this task.

Various models were implemented and evaluated using the quantitative Fréchet Inception Distance (FID) metric and a qualitative custom visual evaluation scheme. The results from the baseline models, CycleGAN, CUT, GcGAN, and StarGAN-v2, indicated that CycleGAN generated the best images based on visual quality for this particular dataset. Based on the metrics implemented, the model reached a visual evaluation score of 3.5 and an FID score of 143.

The results from the various tuning experiments showed a modified CycleGAN model with a ResNet-based architecture in the generator with Xavier weight initialization, ReLU as the non-linear activation, and a learning rate of $1e^{-4}$ to be the best configuration. This modified model generated the most realistic infrared images, reaching a visual evaluation score of 3.75/5 and an FID score of 135. These results indicate a direct improvement to the Specialization Report, where an FID score of 195 was reached. Although the results are promising, with realistic translation for frequent objects in the dataset, such as boats and ferries, it performs worse on less frequent objects, such as kayaks and vessels.

Limitations such as object imbalance, the available amount of images, resolution, and the field-of-view difference between the electro-optical and infrared camera outputs are the dataset's underlying challenges, believed to prohibit the improvement of creating realistic infrared images.

In its current state, the model is not directly suitable for real-world practice due to these limitations. However, the research conducted in this thesis provides a foundation for further improvements and refinements to the unpaired image-to-image translation methods. Addressing the limitations and enhancing the model's performance could become a valuable tool for incorporating infrared cameras into the milliAmpere 2's autonomy system, improving the vessel's situational awareness during nighttime and in poor weather conditions.

The research in the field of unpaired image-to-image translation has made significant progress, mainly in the visual spectrum. There is still room for further exploration across domains, such as from the visible to non-visible (infrared) spectrum. The experiments presented in this thesis could have been further enhanced by exploring additional models, verifying the generalization capabilities of the proposed approach, or customizing evaluation schemes to assess the model's output thoroughly.

6.1 Further Work

To advance in this field, several avenues for future research are proposed, listed in prioritized order:

Improving the Dataset: Collecting a more diverse and representative dataset that includes various types of boats and scene backgrounds could improve model learning. Additionally, ensuring better alignment of images in terms of field of view, perspective, and resolution could enhance the model's translation capabilities.

Model Exploration: Exploring models with more parameters may provide further improvements in translation quality beyond the capabilities of the current CycleGAN model. Utilizing diffusion models, recently made popular [46][34], may be a valid option when applicable code implementations become readily available.

Exploration of Alternative Evaluation methods: To better inform model tuning and optimization efforts, alternative evaluation metrics that are more closely aligned with the nuances of visual performance should be investigated. Appropriately aligning such a metric with image quality would open the possibility for implementing early stopping strategies, which helps reduce overfitting and model degradation during training.

Transfer Learning: Exploring transfer learning techniques to leverage pre-trained models and knowledge from other domains, such as the automotive dataset, Teledyne FLIR ADAS, from FLIR and exploring whether land-based traffic detection transfers appropriately into the maritime environment. This can help reduce the required training time and computational resources while potentially enhancing the system's overall performance.

Automated Model Optimization: Implementing automated hyperparameter optimization techniques, such as grid or random search, could lead to a more systematic discovery of optimal model parameters and improved model performance.

References

[1] Rachael Abbott et al. "Unsupervised object detection via LWIR/RGB translation". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2020, pp. 407–415. DOI: 10.1109/CVPRW50498.2020.00053.

[2] Martín Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein Generative Adversarial Networks". In: *International Conference on Machine Learning*. 2017.

[3] Ali Borji. "Pros and cons of GAN evaluation measures". In: *Computer Vision and Image Understanding* 179 (2019), pp. 41–65. ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2018.10.009. URL: https://www.sciencedirect.com/science/article/pii/S1077314218304272.

[4] Ali Borji. "Pros and cons of GAN evaluation measures: New developments". In: *Computer Vision and Image Understanding* 215 (2022), p. 103329. ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2021.103329. URL: https://www.sciencedirect.com/science/article/pii/S1077314221001685.

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. "Large Scale GAN Training for High Fidelity Natural Image Synthesis". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: https://openreview.net/forum?id=B1xsqj09Fm.

[6] Xi Chen et al. "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets." In: *NIPS*. Ed. by Daniel D. Lee et al. 2016, pp. 2172–2180. URL: http://dblp.uni-trier.de/db/conf/nips/nips2016.html#ChenCDHSSA16.

[7] Y. Choi et al. "StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2018, pp. 8789–8797. DOI: 10.1109/CVPR.2018.00916. URL: https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00916.

[8] Yunjey Choi et al. "StarGAN v2: Diverse Image Synthesis for Multiple Domains". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2020.

[9] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[10] Laurent Dinh, David Krueger, and Yoshua Bengio. "NICE: Non-linear Independent Components Estimation". In: *CoRR* abs/1410.8516 (2014).

[11] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. "Density estimation using Real NVP". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=HkpbnH9lx.

[12] Huan Fu et al. "Geometry-Consistent Generative Adversarial Networks for One-Sided Unsupervised Domain Mapping". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 2422–2431. DOI: 10.1109/CVPR.2019.00253.

[13] Ian Goodfellow et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014. URL: https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

[14] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[15] Kaiming He et al. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1026–1034. DOI: 10.1109/ICCV.2015.123.

[16] Øystein Kaarstad Helgesen et al. "Sensor Combinations in Heterogeneous Multisensor Fusion for Maritime Target Tracking". In: *2019 22th International Conference on Information Fusion (FUSION)*. 2019, pp. 1–9. DOI: 10.23919/FUSION43075.2019.9011297.

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS'20. Vancouver, BC, Canada: Curran Associates Inc., 2020. ISBN: 9781713829546.

[18] P. Isola et al. "Image-to-Image Translation with Conditional Adversarial Networks". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, July 2017, pp. 5967–5976. DOI: 10.1109/CVPR.2017.632. URL: https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.632.

[19] Minguk Kang and Jaesik Park. "ContraGAN: Contrastive Learning for Conditional Image Generation". In: 2020.

[20] Tero Karras, Samuli Laine, and Timo Aila. "A Style-Based Generator Architecture for Generative Adversarial Networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[21] Tero Karras et al. "Alias-Free Generative Adversarial Networks". In: *Proc. NeurIPS*. 2021.

[22] Tero Karras et al. "Analyzing and Improving the Image Quality of StyleGAN". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[23] Tero Karras et al. "Progressive Growing of GANs for Improved Quality, Stability, and Variation". In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=Hk99zCeAb.

[24] Shehryar Khattak, Christos Papachristos, and Kostas Alexis. "Visual-Thermal Landmarks and Inertial Fusion for Navigation in Degraded Visual Environments". In: *2019 IEEE Aerospace Conference*. 2019, pp. 1–9. DOI: 10.1109/AERO.2019.8741787.

[25] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2013. DOI: 10.48550/ARXIV.1312.6114. URL: https://arxiv.org/abs/1312.6114.

[26] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2022. arXiv: 1312.6114 [stat.ML].

[27] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980 (2014).

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[29] Suren Kumar, Tim Marks, and Michael Jones. "Improving Person Tracking Using an Inexpensive Thermal Infrared Sensor". In: June 2014, pp. 217–224. DOI: 10.1109/CVPRW.2014.41.

[30] C. Ledig et al. "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, July 2017, pp. 105–114. DOI: 10.1109/CVPR.2017.19. URL: https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.19.

[31] Rensis Likert. *A technique for the measurement of attitudes / by Rensis Likert.* eng. Archives of psychology ; no. 140. New York: [s.n.], 1985 - 1932.

[32] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. "Unsupervised Image-to-Image Translation Networks". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/dc6a6489640ca02b0d42dabeb8e46bb7-Paper.pdf.

[33] Armin Mehri and Angel Sappa. "Colorizing Near Infrared Images Through a Cyclic Adversarial Approach of Unpaired Samples". In: June 2019, pp. 971–979. DOI: 10.1109/CVPRW.2019.00128.

[34] Midjourney. *Midjourney*. Accessed: 2023-03-30. 2022. URL: https://www.midjourney.com/.

[35] Mehdi Mirza and Simon Osindero. *Conditional Generative Adversarial Nets*. 2014. DOI: 10.48550/ARXIV.1411.1784. URL: https://arxiv.org/abs/1411.1784.

[36] Takeru Miyato et al. "Spectral Normalization for Generative Adversarial Networks". In: Feb. 2018.

[37] Kamyar Nazeri, Eric Ng, and Mehran Ebrahimi. "Image Colorization Using Generative Adversarial Networks". In: *Articulated Motion and Deformable Objects*. Springer International Publishing, 2018, pp. 85–94. DOI: 10.1007/978-3-319-94544-6_9. URL: https://doi.org/10.1007%5C%2F978-3-319-94544-6_9.

[38] Andreas Øie. *Automatic RGB to IR Translation*. Technical Report. Norwegian University of Science and Technology, 2022.

[39] OpenAI. *GPT-4*. Accessed: 2023-03-30. 2023. URL: https://openai.com/research/gpt-4.

[40] OpenAI. *Introducing ChatGPT*. Accessed: 2023-03-30. 2022. URL: https://openai.com/blog/chatgpt.

[41] Yingxue Pang et al. *Image-to-Image Translation: Methods and Applications*. 2021. arXiv: 2101.08629 [cs.CV].

[42] Taesung Park et al. "Contrastive Learning for Conditional Image Synthesis". In: *ECCV*. 2020.

[43] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[44] Alec Radford, Luke Metz, and Soumith Chintala. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. 2015. DOI: 10.48550/ARXIV.1511.06434. URL: https://arxiv.org/abs/1511.06434.

[45] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, 22–24 Jun 2014, pp. 1278–1286. URL: https://proceedings.mlr.press/v32/rezende14.html.

[46] Robin Rombach et al. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2021. arXiv: 2112.10752 [cs.CV].

[47] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab et al. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4.

[48] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning Internal Representations by Error Propagation". In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*. Ed. by David E. Rumelhart and James L. Mcclelland. Cambridge, MA: MIT Press, 1986, pp. 318–362.

[49] Jascha Sohl-Dickstein et al. "Deep Unsupervised Learning using Nonequilibrium Thermodynamics". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 2256–2265.

[50] Yang Song and Stefano Ermon. "Generative Modeling by Estimating Gradients of the Data Distribution". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/3001ef257407d5a371a96dcd947c7d93-Paper.pdf.

[51] Christian Szegedy et al. "Rethinking the Inception Architecture for Computer Vision". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2818–2826. DOI: 10.1109/CVPR.2016.308.

[52] Weilun Wang et al. "Instance-wise Hard Negative Example Generation for Contrastive Learning in Unpaired Image-to-Image Translation". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 14000–14009. DOI: 10.1109/ICCV48922.2021.01376.

[53] Shihao Yang et al. "An Unpaired Thermal Infrared Image Translation Method Using GMA-CycleGAN". In: *Remote Sensing* 15.3 (2023). ISSN: 2072-4292. DOI: 10.3390/rs15030663. URL: https://www.mdpi.com/2072-4292/15/3/663.

[54] Maoxun Yuan, Yinyan Wang, and Xingxing Wei. "Translation, Scale and Rotation: Cross-Modal Alignment Meets RGB-Infrared Vehicle Detection". In: *Computer Vision – ECCV 2022*. Ed. by Shai Avidan et al. Cham: Springer Nature Switzerland, 2022, pp. 509–525.

[55] Syed Waqas Zamir et al. "Multi-Stage Progressive Image Restoration". In: *CVPR*. 2021.

[56] Syed Waqas Zamir et al. *Restormer: Efficient Transformer for High-Resolution Image Restoration*. 2021. DOI: 10.48550/ARXIV.2111.09881. URL: https://arxiv.org/abs/2111.09881.

[57] He Zhang, Vishwanath Sindagi, and Vishal M. Patel. *Image De-raining Using a Conditional Generative Adversarial Network*. 2017. DOI: 10.48550/ARXIV.1701.05957. URL: https://arxiv.org/abs/1701.05957.

[58] Jian Zhao and Sen-ching S. Cheung. "Human segmentation by fusing visible-light and thermal imaginary". In: *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*. 2009, pp. 1185–1192. DOI: 10.1109/ICCVW.2009.5457476.

[59] Jun-Yan Zhu et al. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks". In: *Computer Vision (ICCV), 2017 IEEE International Conference on*. 2017.

Appendices

A Training Scripts

Various training scripts used for training each baseline model.

CycleGAN

```
python train.py \
--dataroot /home/andreoi/data/autoferry \
--name autoferry_cyc \
--model cycle_gan \
--input_nc 3 \
--output_nc 1 \
--batch_size 8 \
--phase train \
--save_by_iter \
--lambda_identity 0.0 \
--preprocess resize \
--load_size 256
```

Original implementation: *https://github.com/taesungp/contrastive-unpaired-translation*
Thesis implementation: *https://github.com/andreasoie/cutslim*

CUT

```
python train.py \
--dataroot /home/andreoi/data/autoferry \
--name autoferry \
--model cut \
--input_nc 3 \
--output_nc 3 \
--batch_size 8 \
--phase train \
--save_by_iter \
--preprocess resize \
--load_size 256
```

Original implementation: *https://github.com/taesungp/contrastive-unpaired-translation*
Thesis implementation: *https://github.com/andreasoie/cut*

GcGAN

```
python train.py
--dataroot /home/andreoi/data/autoferry \
--name rgb2ir \
--phase train \
--model gc_gan_cross \
--pool_size 50 \
--no_dropout \
--loadSize 256 \
--which_model_netG resnet_6blocks \
--which_direction AtoB \
--dataset_mode unaligned \
--resize_or_crop resize \
--batchSize 8 \
--nThreads 4 \
--input_nc 3 \
--output_nc 3 \
--identity 0.3 \
--geometry rot \
--niter 200 \
--niter_decay 200
```

Original implementation: *https://github.com/hufu6371/gcgan*
Thesis implementation: *https://github.com/andreasoie/gcgan*

StarGAN-v2

```
python main.py \
--mode train \
--num_domains 2 \
--w_hpf 0 \
--lambda_reg 1 \
--lambda_sty 1 \
--lambda_ds 1 \
--lambda_cyc 1 \
--train_img_dir data/autoferry/train \
--val_img_dir data/autoferry/val \
--total_iters 71000
```

Original implementation: *https://github.com/clovaai/stargan-v2*
Thesis implementation: *https://github.com/andreasoie/starganv2*

B Network Capacities for the Baseline models

CycleGAN

netG_A	netG_B	netD_A	netD_B
11.37M	11.37M	2.76M	2.77M

Table B.0.1: The number of parameters for the CycleGAN networks

CUT

netG	netF	netD
11.38M	0.56M	2.77M

Table B.0.2: The number of parameters for the CUT networks

GcGAN

netG_AB	netG_gc_AB	netD_B	netD_gc_B
7.84M	7.84M	2.76M	2.76M

Table B.0.3: The number of parameters for the GcGAN networks

StarGAN-v2

netG	netM	SE	netD
33.89M	2.44M	20.92M	20.85M

Table B.0.4: The number of parameters for the StarGAN-v2 networks

C Code Snippets

Calculating Misalignment

```python
import numpy as np

def get_bbox_info(bbox):
    xmin, ymin, xmax, ymax = bbox
    width = xmax - xmin
    height = ymax - ymin
    center_x = xmin + width / 2
    center_y = ymin + height / 2
    angle = np.arctan2(height, width)

    return center_x, center_y, width, height, angle

def compare_bboxes(bbox1, bbox2):
    center_x1, center_y1, width1, height1, angle1 = get_bbox_info(bbox1)
    center_x2, center_y2, width2, height2, angle2 = get_bbox_info(bbox2)

    position_deviation = np.sqrt((center_x2 - center_x1)**2 + (center_y2 - center_y1)**2)
    scale_deviation = np.sqrt((width2 - width1)**2 + (height2 - height1)**2)
    rotation_deviation = np.abs(angle2 - angle1)

    return position_deviation, scale_deviation, rotation_deviation
```

Baseline Configuration

```python
baseline_config = {
    'batch_size': 8,
    'batch_size_fid': 9,
    'beta1': 0.5,
    'beta2': 0.999,
    'checkpoints_dir': 'checkpoints',
    'continue_train': False,
    'crop_size': 256,
    'dataroot': '/home/andreoi/data/autoferry',
    'dataset_mode': 'unaligned',
    'direction': 'AtoB',
    'epoch': 0,
    'epoch_count': 1,
    'freq_display': 400,
    'freq_log': 100,
    'freq_save_epoch': 25,
```

```
    'gan_mode': 'lsgan',
    'gpu_ids': [0],
    'init_gain': 0.02,
    'init_type': 'xavier',
    'input_nc': 3,
    'isTrain': True,
    'lambda_A': 10,
    'lambda_B': 10,
    'lambda_identity': 0,
    'load_size': 256,
    'lr': 0.0002,
    'lr_decay_iters': 50,
    'lr_policy': 'linear',
    'max_dataset_size': inf,
    'model': 'cycle_gan',
    'n_epochs': 200,
    'n_epochs_decay': 200,
    'n_layers_D': 3,
    'name': 'autoferry_cycle_gan',
    'ndf': 64,
    'netD': 'basic',
    'netG': 'resnet_9blocks',
    'ngf': 64,
    'no_antialias': False,
    'no_antialias_up': False,
    'no_dropout': True,
    'no_flip': False,
    'normD': 'instance',
    'normG': 'instance',
    'num_threads': 4,
    'outdir': 'outputs',
    'output_nc': 1,
    'phase': 'train',
    'pool_size': 50,
    'preprocess': 'resize',
    'pretrained_name': None,
    'random_scale_max': 3,
    'save_by_iter': False,
    'serial_batches': False,
    'shuffle': True,
    'stylegan2_G_num_downsampling': 1,
    'suffix': '',
    'verbose': False,
    'wandb': True
}
```

D Web Application: Annotating Images

Created a web appliation for manually classifying images

The application can be found in `https://github.com/andreasoie/imagelabeler`. It contains both the UI and the API required to run the image-labeling application.

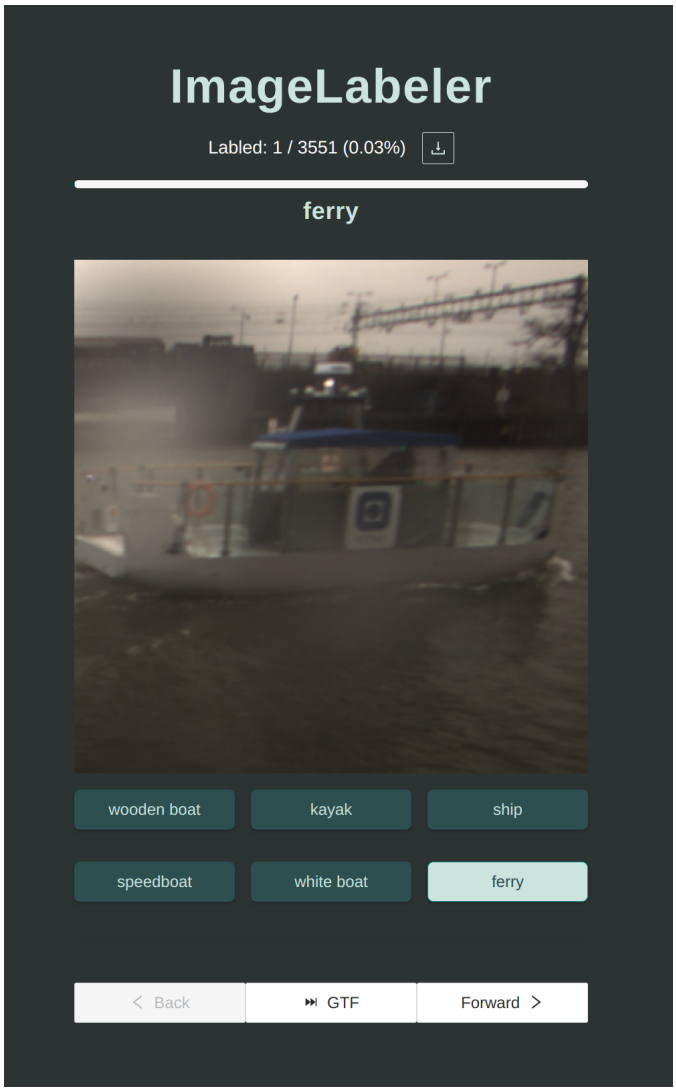


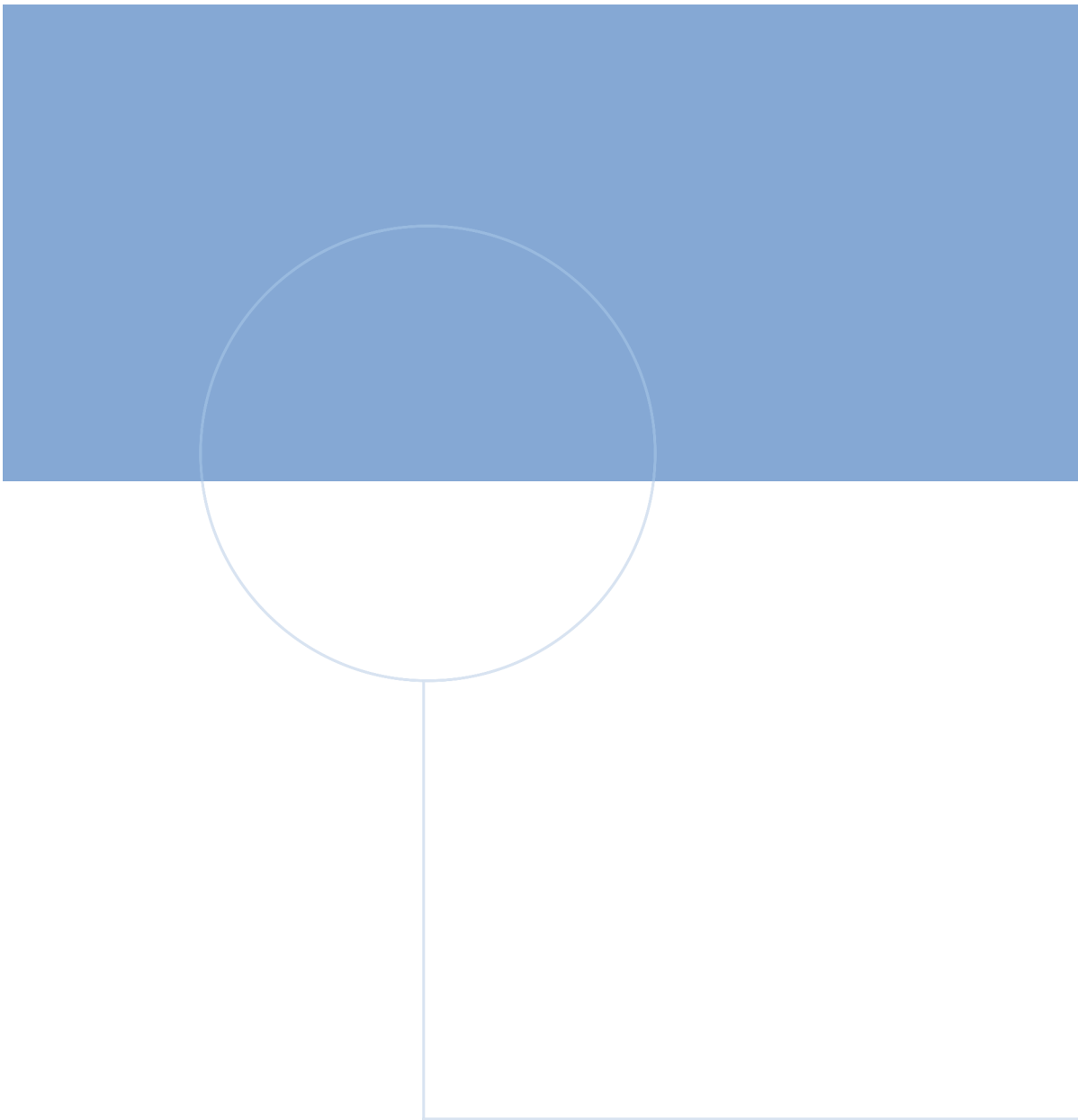
Figure .1: The user-interface of the image-labeling application. Image shows the rescaled image-crop from the dataset with several labels to chose between.

E Example Video

Time-lapse of generated class-objected using the Final Model

The video shows various generated infrared image outputs during training. The images were sampled every 400 iteration (as stated in the Baseline Configuration from Appendix C).

Video: https://www.youtube.com/watch?v=pmbXAHC8r3o



NTNU

Norwegian University of
Science and Technology