Technische Hochschule Deggendorf

Fakultät Angewandte Informatik

Studiengang Master Angewandte Informatik

Künstliche Intelligenz-gesteuerte Bildsynthese für Sensordaten

Artificial Intelligence-driven Image Synthesis for Sensor Data

Masterarbeit zur Erlangung des akademischen Grades:

Master of Science (M.Sc.)

an der Technischen Hochschule Deggendorf

Vorgelegt von:

Darshan Hassan Shivakumar

Matrikelnummer: 12300360

Am: 30.01.2026

Prüfungsleitung:

Stefan Kunze, M.Sc.

Bibliographic details:

Darshan, H., 2026: Artificial Intelligence-driven Image Synthesis for Sensor Data. Mgr. Thesis, in English. – 83 p., Faculty of Science, University of South Bohemia, České Budějovice, Czech Republic.

Annotation:

This thesis investigates generative models and identifies a suitable model for each sensor modality (RGB camera, IR sensor, radar, and laser scanner) using data collected from drone-mounted sensors. The goal is to generate high-quality synthetic images that are difficult to acquire at large scale due to limited flight time, regulatory restrictions, safety concerns, and environmental constraints. A common methodological workflow is defined for all sensor modalities, and the model selection and the evaluation strategies are provided for each modality. However, experimental implementation is conducted only for images collected by the RGB camera.

Declaration:

I declare that I am the author of this qualification thesis and that in writing it I have used the sources and literature displayed in the list of used sources only.

I further declare that I have used the following generative artificial intelligence tools and services in accordance with academic ethics:

- ChatGPT (OpenAI)
- Gemini (Google DeepMind)
- Grammarly

for the purpose of identification of relevant research literature, conceptual clarification, debugging, grammar correction and refinement.

České Budějovice, 30.01.2026

Acknowledgement

With a strong interest in Generative Artificial Intelligence, I gladly accepted the thesis opportunity offered by Technology Campus Freyung (Research Institute of "Deggendorf Institute of Technology"). It deepened my understanding of the generative models for different types of sensor images. I would like to express my gratitude to Stefan Kunze, M.Sc., for the guidance and feedback throughout the research work. His suggestions played an important role in the writing of this research work.

I would also like to thank Simon Uhrman, M.Sc., and Mittalben Panchal, M. Eng, for their support, discussions, and feedback, which contributed to both the technical and conceptual development of the work. Finally, I would like to acknowledge the technical support received from Technology Campus Freyung (TCF), which enabled me to conduct this research effectively.

Contents

1 Introduction 1
1.1 Motivation 1
1.2 Problem Statement 1
1.3 Research Questions 2
1.4 Research Objectives 2
1.5 Research Methodology 2
1.6 Sensor Data Characteristics 3
1.7 Outline 5
2 State of the Art 7
2.1 Generative AI 7
2.2 Generative Models 7
2.2.1 Generative Adversarial Networks (GANs) 8
2.2.2 Diffusion Models (DMs) 9
2.2.3 Variational Autoencoders 11
3 Literature Review 13
3.1 RGB-Camera Data Synthesis 13
3.2 IR-Camera Data Synthesis 20
3.3 RADAR Sensor Data Synthesis 26
3.4 Laser Scanner Data Synthesis 30
4 Methodology 33
4.1 Conceptual Framework 33
4.2 RGB Camera Sensor Data 35
4.2.1 Model Selection 36
4.2.2 Data Acquisition and Preparation 40
4.2.3 Training 41
4.2.4 Evaluation 42
4.3 IR Camera Sensor Data 43
4.3.1 Model Selection 44
4.3.2 Data Acquisition and Preparation 47

4.3.3 Model Training 48
4.3.4 Evaluation 49
4.4 RADAR Sensor Data 51
4.4.1 Model Selection 51
4.4.2 Dataset Preparation 54
4.4.3 Model Training 55
4.4.4 Evaluation 56
4.5 Laser Scanner Sensor Data 57
4.5.1 Model Selection 57
4.5.2 Dataset Preparation 60
4.5.3 Model Training 60
4.5.4 Evaluation 61
5 Results 62
5.1 RGB Camera Sensor Data 62
5.2 Quantitative Evaluation 62
5.3 Qualitative Evaluation 66
5.4 Model Performance Discussion 75
5.5 Limitations 76
6 Conclusion and Future Scope 78
6.1 Conclusion 78
6.2 Future Scope 79
7 Bibliography 80

List Of Acronyms

2D – Two-Dimensional

3D – Three-Dimensional

ADAS V2 – Advanced Driver Assistance Systems Dataset Version 2

AI – Artificial Intelligence

APE – Absolute Positional Embeddings

BEV – Bird's-Eye View

CAE – Convolutional Autoencoder

CGAN – Conditional Generative Adversarial Network

CNN – Convolutional Neural Network

CVPR – Conference on Computer Vision and Pattern Recognition

CUT – Contrastive Unpaired Translation

DCGAN – Deep Convolutional Generative Adversarial Network

DDPM – Denoising Diffusion Probabilistic Model

DM – Diffusion Model

DUSty – Diffusion Using Sparse Trajectories

FFT – Fast Fourier Transform

FID – Fréchet Inception Distance

FMCW – Frequency-Modulated Continuous-Wave

FPD – Fréchet Point Cloud Distance

FPVD – Fréchet Point-Based Volume Distance

FRD – Fréchet Range Distance

FRID – Fréchet Range Image Distance

FSVD – Fréchet Sparse Volume Distance

GAN – Generative Adversarial Network

HDiT – Hourglass Diffusion Transformer

HSV – Hue, Saturation, and Value

ICRA – International Conference on Robotics and Automation

IF – Intermediate Frequency

IQ – In-Phase and Quadrature

IR – Infrared

IS – Inception Score

JSD – Jensen–Shannon Divergence

KITTI-360 – Karlsruhe Institute of Technology and Toyota Technological Institute 360 Dataset

LiDAR – Light Detection and Ranging

LiDM – LiDAR Diffusion Model

LIDARGen – LiDAR Generation Model

LLM – Large Language Model

LPIPS – Learned Perceptual Image Patch Similarity

LSUN – Large-scale Scene Understanding

M3FD – Multi-Spectral, Multi-Scene, and Multi-Resolution Fusion Dataset

MMD – Maximum Mean Discrepancy

MTI – Moving Target Indicator

NFE – Number of Function Evaluations

ODE – Ordinary Differential Equation

PSNR – Peak Signal-to-Noise Ratio

R2DM – Range-Reflectance Diffusion Model

R2Flow – Range-Reflectance Flow

RA – Range-Azimuth

RADAR – Radio Detection and Ranging

RD – Range-Doppler

RF – Rectified Flow

RGB – Red, Green, and Blue

SIFID – Single Image Fréchet Inception Distance

SSIM – Structural Similarity Index Measure

TD – Timestep Distillation

UAV – Unmanned Aerial Vehicle

U-Net – U-shaped Neural Network

VAE – Variational Autoencoder

ViT – Vision Transformer

WGAN – Wasserstein Generative Adversarial Network

List of Figures

Figure 1: Example of RD map [4] 4
Figure 2: Example of RA map [4] 5
Figure 3: GAN Architecture based on [8] 8
Figure 4: The Forward Diffusion Process based on [17] 10
Figure 5: The Reverse Diffusion Process [17] 10
Figure 6: Variational Auto Encoder (VAE) architecture [8] 11
Figure 7: SinDiffusion Denoising Network based on [25] 18
Figure 8: Algorithm Structure of IR-GAN [31] 25
Figure 9: The CGAN architecture used [10] 29
Figure 10: Synthetic Data Generation Process for Sensor data 34
Figure 11: Algorithm processing chain based on [4] 55
Figure 12: Downward Facing (DF) Camera Images 63
Figure 13: Forward Facing (FF) Camera Images 63
Figure 14: Real Samples from Place50 Dataset [44] 64
Figure 15: Best Fidelity Image from DF generation (SIFID = 0.21) 67
Figure 16: Worst Fidelity Image from DF generation (SIFID = 1.16) 67
Figure 17: Least Diverse Image Pairs from DF generation (LPIPS = 0.15) 68
Figure 18: Most Diverse Image Pairs from DF generation (LPIPS = 0.80) 69
Figure 19: Best Fidelity Image from FF Generation (SIFID = 0.08) 69
Figure 20: Worst Fidelity for FF Generation (SIFID = 0.61) 70
Figure 21: Least Diverse Image Pairs for FF Generation (LPIPS = 0.18) 71
Figure 22: Most Diverse Image Pairs from FF generation (LPIPS = 0.70) 72
Figure 23: Generated samples for selected DF, FF, and P50 scenes. 75

List of Tables

Table 1: Strengths and Weaknesses of GANs and Stable Diffusion [23] 14
Table 2: Ranking of the methods for RGB image Synthesis 38
Table 3: Summary of other contenders 40
Table 4: Tunable Hyperparameters for SinDiffusion Training 42
Table 5: Ranking of the methods for Thermal image Synthesis 45
Table 6: Summary of other contenders 47
Table 7: Tunable Hyperparameters for IR-GAN 49
Table 8: Comparison between Generator-Predictor system and CGAN 53
Table 9: Ranking of the models for Laser Scanner Image Synthesis 58
Table 10: Summary of other contenders 59
Table 11: Quantitative Results for RGB image synthesis 65

1 Introduction

1.1 Motivation

Synthetic data generation for multi-modal sensors is essential because it can provide large amounts of diverse training samples that are difficult to acquire due to cost, time and hardware limitations involved in collecting real-world data. It improves the accuracy and reliability of deep multi-modal perception systems for autonomous driving in adverse weather conditions and in scenarios involving sensor degradation [1].
The acquisition of a high number of real-world data samples from multi-modal sensors has certain limitations and drawbacks due to their physical experiments, which require expensive hardware and lot of labeling work. The researchers have identified an effective method to overcome these limitations, which is through the creation of synthetic datasets [2]. The drones are equipped with various sensor modalities such as RGB-Cameras, Infrared (IR) Cameras, LIDAR sensors, and RADAR sensors. This enables multi-modal data acquisition in applications such as agriculture, forestry and natural environment [3]. However, collecting large-scale and diverse UAV data is a challenge due to high operational costs, extreme weather conditions (e.g., snow/fog/heavy rain/strong winds), limited flight time, regulatory restrictions and safety concerns. The synthetic image generation therefore provides a solution for producing scalable and diverse images for training and evaluating object detection models.

1.2 Problem Statement

The aim of this thesis is to investigate generative models for generating synthetic images using the data collected from drone-mounted multi-modal sensors in real-world environments. The sensors considered in this research include RGB camera, IR camera, RADAR sensor, and a laser scanner. The generative models are used to produce synthetic images but how well the generated images replicate real images in natural environments for different sensor modalities remains unclear. This research investigates which existing generative model is best suitable for producing high-quality and diverse synthetic images

for each sensor modality through methodological discussion. However, the experimental evaluation is performed only for the collected RGB images.

1.3 Research Questions

1. Which generative model is most suitable for generating synthetic images across each sensor modality (RGB, IR, RADAR, and laser scanner)?
2. How is the quality of synthesized images evaluated for each sensor modality?
3. How efficiently does the selected generative model perform for RGB image synthesis?

1.4 Research Objectives

This research is focused on analysing the generative models for synthetic image generation across drone-mounted sensor modalities, which includes an RGB camera, an IR camera, a RADAR sensor, and a laser scanner. A methodological workflow is designed for generating synthetic images from sensor data and the most suitable generative model for each modality will be identified based on the literature and the data characteristics of the sensor. Then the evaluation strategies are discussed to assess the quality of generated synthetic images across modalities. Finally, the selected model will be implemented for drone-acquired RGB images, and the synthesized images will be evaluated to assess model performance.

1.5 Research Methodology

This thesis investigates generative techniques for synthesizing images using data collected from drone-mounted sensors. It begins with a detailed review of existing literature to understand how different generative AI approaches are applied in the synthetic image generation and uses these findings to select suitable generative models for each sensor modality. Based on the reviewed research, appropriate evaluation strategies are then defined for each modality. The experimental implementation is carried out using only the collected images from the RGB camera sensor, where the selected generative model is

trained on real RGB images and the synthesized images are evaluated through quantitative metrics as well as qualitative analysis.

1.6 Sensor Data Characteristics

The drone-mounted sensor data used for synthetic image generation across different sensor modalities are discussed in this section. The drone equipped with sensors such as an RGB camera, an infrared (IR) camera, a RADAR sensor, and a laser scanner have distinct data characteristics, and their details are given below

RGB-Camera Data:

The RGB camera sensor data is acquired using drone DJI Mavic 3M (Multispectral) RGB camera. The drone system includes both forward-facing and downward-facing RGB cameras that produce high-resolution two-dimensional color images having three channels Red, Green, and Blue. The images captured by the downward-facing camera consist of ground surfaces and the forward-facing camera provides the surrounding landscapes and terrain views. The resulting images from these sensors are stored in JPEG format.

IR-Camera Data:

The infrared camera sensor data is obtained using DJI Matrice 30T drone system equipped with a wide RGB camera and an infrared (IR) thermal sensor. This allows for the simultaneous capture of RGB and thermal data of the same scene. The infrared images contain information about the temperature of the objects. The sensor output consists of synchronous video streams stored in .MP4 format. The files designated with the suffix _W correspond to the wide RGB camera video stream, while files with the suffix _S correspond to the IR thermal video stream.

RADAR Sensor Data:

The data is acquired using a BGT60TR13C Frequency Modulated Continuous Wave (FMCW) radar sensor. The data can be represented as two-dimensional Range-Doppler (RD) map or two-dimensional Range-Angle (RA) map [4].

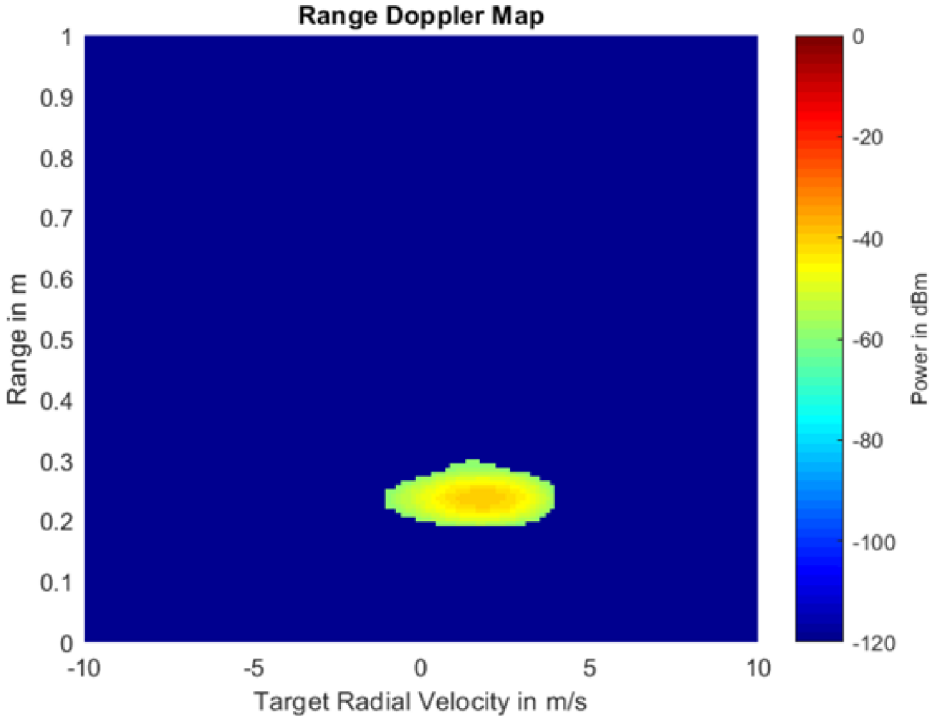


Figure 1: Example of RD map [4]

Figure 1 is an example of RD map. It's a two-dimensional grid representation, which visualizes the ranges and velocities of the detected objects. The y-axis corresponds to the range (distance) of the target and x-axis corresponds to the velocity of the target object. The positive velocity values indicate that the object is moving away from the sensor, while negative velocity values indicate object approaching towards the radar. The strength of the detected objects is represented by varying colours within a colormap [4].

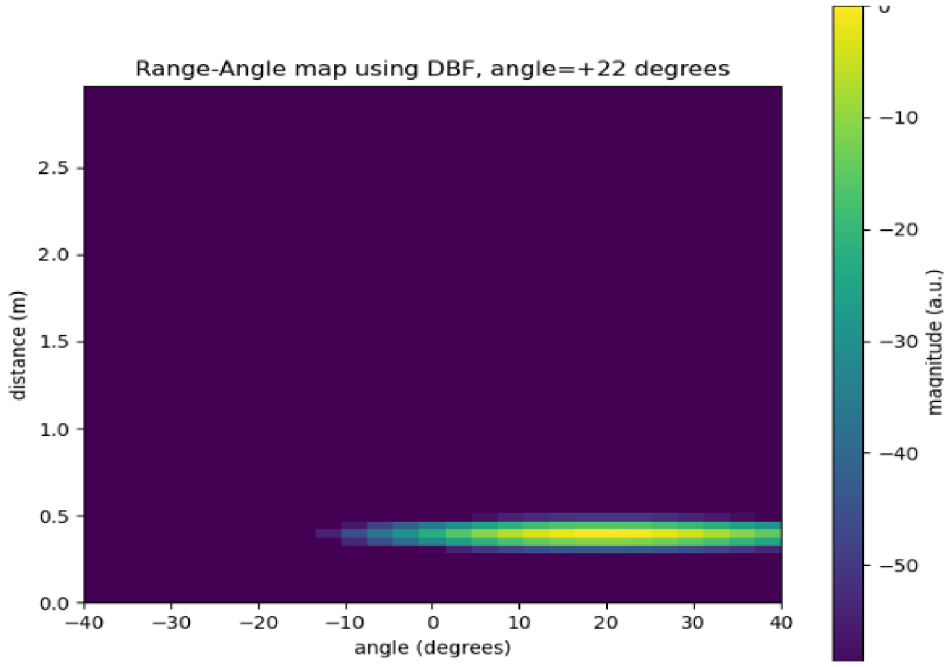


Figure 2: Example of RA map [4]

Figure 2 is an example of RA map. It illustrates the range and angle of the detected objects relative to the radar system. The plot represents the signal magnitude (in arbitrary units) of the detected objects in terms of angle (in degrees) and distance (in meters), where the brighter colors indicate stronger signal intensities [4].

Laser Scanner Data:

The laser scanner data is acquired using YellowScan Mapper, a UAV-based LiDAR system. This sensor outputs data in the LAS file format. The data consists of three-dimensional point clouds, where individual points are defined by their three-dimensional coordinates (X, Y, and Z) along with a reflectance value representing the strength of the laser return signal [5].

1.7 Outline

The remainder of this thesis is organised as follows. Chapter 2 provides an overview of Generative Artificial Intelligence and reviews relevant generative models. Chapter 3 presents a literature review for each sensor modality and discusses the models used to

address the research problem. Chapter 4 describes the research methodology, including the workflow for synthetic image generation, sensor-data characteristics, the selection of generative models, and the evaluation strategy. Chapter 5 presents the experimental results, analyses the generated images, and discusses limitations. Chapter 6 concludes the thesis, summarises the findings, and outlines future work. Finally, Chapter 7 provides the bibliographic information.

2 State of the Art

2.1 Generative AI

Generative AI refers to computational methods that can produce new and meaningful data such as text, images, or audio using training data. It can mimic the real data and assist humans in answering questions and generate helpful insights [6]. The generative AI for text has shown impressive abilities in natural language processing, especially in text generation, summarization, answering questions, and language translation. Large language models (LLMs) like GPT, Llama, Gemini and DeepSeek are making significant improvements in these tasks. The generative AI for image synthesis has challenges because visual data is high-dimensional, complex, and scenes vary endlessly, which makes it difficult to generate realistic images. However, the progress continues in building generative models for image synthesis. Generative AI creates synthetic data that resembles the training data and has found applications in computer vision, natural language processing, music composition, and drug discovery. In computer vision, the Generative AI enhances data augmentation by synthesizing realistic images [7]. Generative AI consists of several generative models for the synthesis of images, and among the most prominent generative models are Generative Adversarial Networks (GANs), Diffusion Models (DMs), and Variational Autoencoders (VAEs). Each of these architectures provides distinct methods for learning data representations and producing new data, which makes them valuable tools for image generation tasks [8].

2.2 Generative Models

This section provides the fundamental architectures used in the generation of synthetic images. The trained generative model learns the underlying data distribution of the real images and produces new images that resemble the original images. These architectures are the backbone of the synthetic image generation, and they are particularly important when there is limited data available in the real-world due to difficulty in acquiring or due to cost. The section focuses on three dominant generative modeling frameworks, which

include Generative Adversarial Networks (GANs), Diffusion Models, and Variational Autoencoders (VAEs).

2.2.1 Generative Adversarial Networks (GANs)

Generative Adversarial Networks are introduced by Goodfellow et al. in 2014 [9]. Since its introduction, they have been in research and praised for their innovative approach in computer vision and image processing tasks [8].

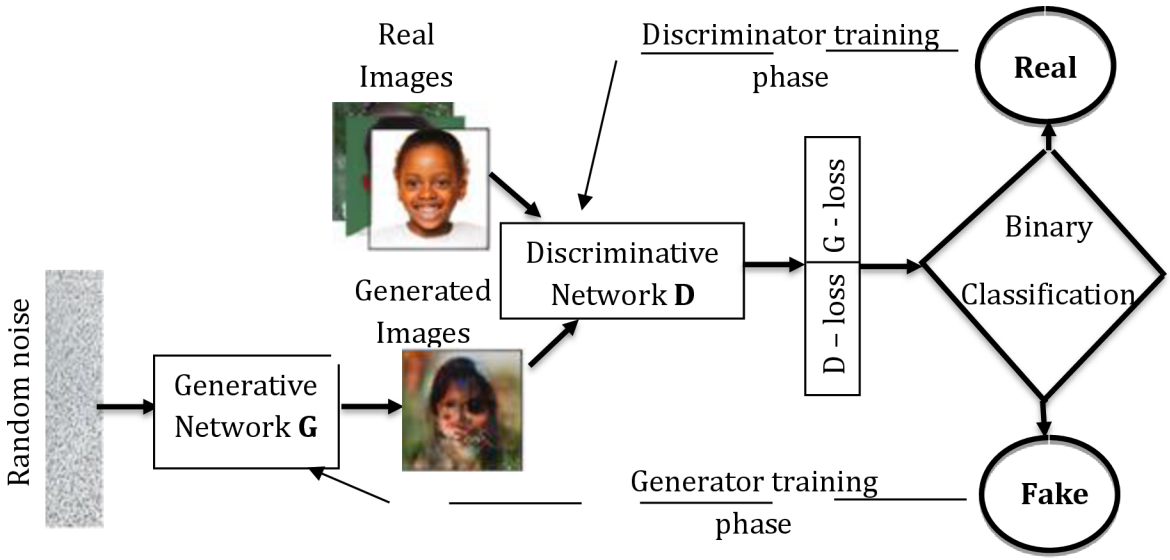


Figure 3: GAN Architecture based on [8]

Figure 3 illustrates GAN, which consists of two Neural Networks (NNs): the Generator network G and the Discriminator network D. The generator learns to synthesize synthetic images that resemble real images, while the discriminator evaluates the generated images to determine whether they are real or fake. The goal of the generator is to synthesize images that look real, while the goal of the discriminator is to differentiate between real and fake image correctly. The adversarial process between the generator and the discriminator network helps the GAN to improve iteratively, which enhances the quality of the generated images to look real over time [8]. The process begins by feeding random noise into the

generator, which creates synthetic images. The generated images and the real images from a training dataset are passed to the discriminator. The discriminator tries to distinguish between real and fake images correctly. During training, the discriminator learns to become better at identifying real versus fake images, while the generator learns to produce more convincing images that can fool the discriminator. Both networks are optimized using two types of loss function, where the discriminator loss (D-loss) which measures how well the discriminator differentiates between real and fake images, and the generator loss (G-loss) measures how well the generator fools the discriminator. This adversarial process helps GANs to improve iteratively, thus enhancing the realism and quality of the generated images over time [8], [9], [10]. Different GAN variants include Conditional GAN [11], Deep Convolutional GAN [12], CycleGANs [13], StyleGANs [14], and Super Resolution GANs [15], which enable effective solutions across diverse applications. The frameworks are flexible which allows the use of different generator and discriminator architectures [8]. GANs can generate high-quality images and can achieve strong generative effectiveness with high inference speed [16]. GANs training can be unstable, which can cause mode collapse, and result in the generator producing similar images to try and fool the discriminator. GANs require a considerable number of computational resources for generating high-resolution images, which makes training expensive and time-consuming. They may overfit the training data, which limits the diversity in the generated images. The evaluation of GANs generated images is challenging, because there are limited reliable metrics available for evaluation [8].

2.2.2 Diffusion Models (DMs)

Diffusion Models (DMs) are probabilistic generative models that synthesize new images by learning to reverse a gradual noise corruption process. They define a forward diffusion process that incrementally adds small amounts of Gaussian noise to real images over a sequence of timesteps, and a learned reverse denoising process that iteratively removes noise to obtain clean new images from the data distribution [17].

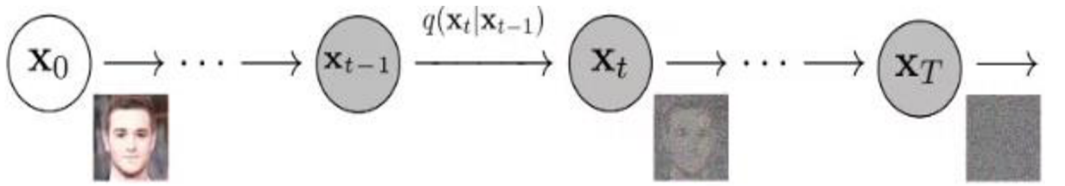


Figure 4: The Forward Diffusion Process based on [17]

The diffusion process consists of two phases: a forward diffusion (noising) phase and a reverse (denoising) phase. Figure 4 illustrates the forward diffusion process, where the noise is gradually introduced into an image until the sample becomes approximately standard Gaussian noise. In the forward phase, a Markov chain is defined by a transition distribution $q(x_t|x_{t-1})$, which incrementally adds Gaussian noise according to predefined noise schedule. After T steps, the resulting image x_T is a pure noisy image [16], [17].

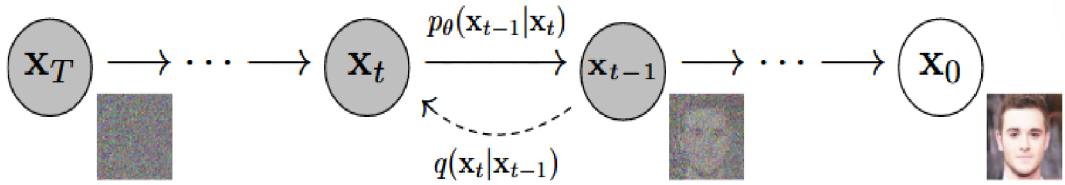


Figure 5: The Reverse Diffusion Process [17]

In the reverse diffusion process, the model learns the reverse-time transitions $p_\theta(x_{t-1}|x_t)$, where θ denotes denoising network learnable parameters. Figure 5 illustrates the reverse phase where model learns the reverse-time transitions of this Markov chain, which enable it to iteratively remove noise from the noisy sample and thereby generate new data samples that match the training distribution [17]. The Diffusion model variants include Denoising Diffusion Probabilistic Models (DDPMs) [17], Improved DDPMs [18], and Latent Diffusion Models [19], which enable high-quality and stable image synthesis across diverse applications.

2.2.3 Variational Autoencoders

Variational Autoencoders (VAEs) are generative models proposed by Kingma and Welling [20]. They are a popular neural network architecture designed for unsupervised learning of complex data distributions [8].

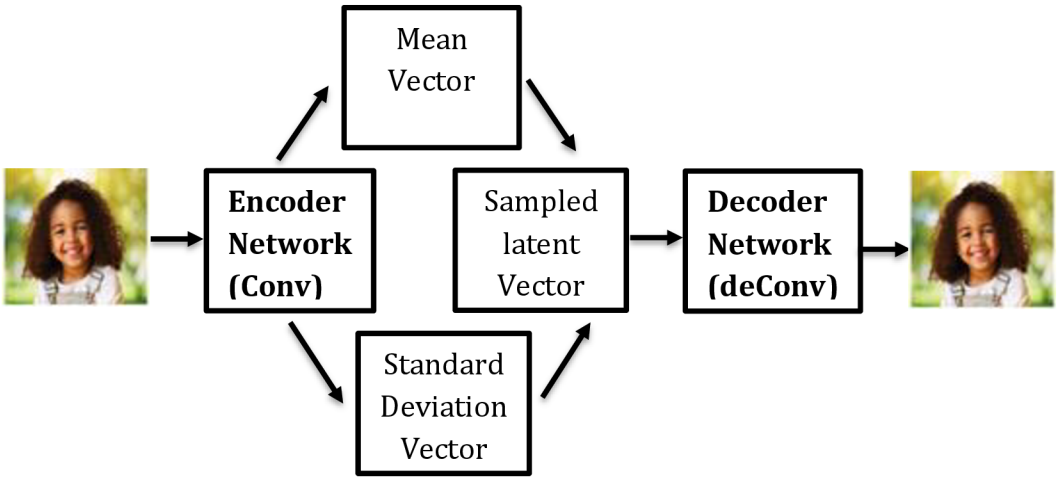


Figure 6: Variational Auto Encoder (VAE) architecture [8]

Figure 6 presents the architecture of VAE, which consists of two main components, an encoder and a decoder. The input image is transformed into a lower-dimensional latent space by the encoder, which creates a latent representation by generating key parameters like mean and variance of a Gaussian distribution. A latent vector is then sampled from this distribution using reparameterization trick and fed into decoder, which reconstructs the original image. The encoder and decoder architectures can be implemented by either using fully connected or convolutional neural networks. During training, the VAE learns meaningful latent representations, which enables effective reconstruction and generative capability [8], [20]. The Variational Autoencoder (VAE) variants include the standard Variational Autoencoder (VAE) [20], β-Variational Autoencoders (β-VAEs) [21], and Conditional Variational Autoencoders (CVAEs) [22], which enable efficient learning of latent representations and generative modelling of complex data distributions across diverse applications. They provide an efficient representation of the underlying structure in a lower-dimensional latent space and are highly effective for generative modelling tasks,

which enable the learning of complex probability distributions from data. However, VAEs often struggle to capture fine details, which results in blurry or fuzzy reconstructions that smooth out high-frequency details. They face significant challenges such as balancing reconstruction accuracy with latent space regularization and susceptibility to posterior collapse, where the model disregards the latent variables, leading to less diverse and meaningless samples [8].

3 Literature Review

Chapter 3 reviews state-of-the-art models in synthetic image generation for the relevant sensor modalities. It covers generative models for image synthesis using data obtained from RGB cameras, IR cameras, RADAR sensors, and LiDAR sensors. The literature is analyzed for each sensor modality independently to understand how the generative models handle sensor data characteristics and how the previous work evaluates the synthesized images. The review discusses RGB image synthesis approaches, thermal image synthesis using translation approaches, RADAR data synthesis, and LiDAR data synthesis.

3.1 RGB-Camera Data Synthesis

In 2024, Gamal (Sadek) et al. published a paper in the "International Journal of Telecommunications" to synthesize images using Stable Diffusion and GANs [23]. The primary focus was to investigate the generative techniques and their applications. The authors describe key features of GAN variants such as DCGAN, WGAN, StyleGAN, CycleGAN, and BigGAN, and they address the limitations such as training instability and mode collapse. Stable Diffusion is a diffusion-based approach that generates data through the gradual reversal of added noise through an iterative denoising process. Stable Diffusion does not rely on adversarial loss, which makes it less prone to mode collapse and more stable during training. The diffusion-based models generate images with high diversity and fewer artifacts, but due to the iterative denoising process it results in slower generation and needs high computational requirements. GANs and diffusion models were trained on benchmark datasets such as CIFAR-10, CelebA, and ImageNet. The experimental results of the trained GAN models were evaluated using IS (quality and diversity) and FID (quality) metrics. The results showed that StyleGAN and BigGAN achieve strong quality metrics, StyleGAN on CelebA scored IS 9.4 and FID 12.5 with minimal mode collapse, and BigGAN on ImageNet scored IS 9.2 and FID 15.6 with no observed mode collapse, while DCGAN on CIFAR-10 scored IS 7.8, FID 31.2 with mode collapse problem and indicated reduced diversity. The diffusion models include Stable Diffusion, DDPM, Improved DDPM, and Vanilla Diffusion. The experimental results indicate improved diversity and competitive

quality metrics relative to the GANs. Improved DDPM on CelebA scored IS 9.6 and FID 9.3 with strong image quality but required longer training time (48 Hours) due to increased denoising steps (2000), Stable Diffusion on CIFAR-10 scored IS 8.1 and FID 19.4 with competitive image quality (36 Hours, 1000 steps), DDPM on CIFAR-10 scored IS 8.5 and FID 20.2 (40 Hours, 1500 steps), and Vanilla Diffusion on ImageNet scored IS 8.7 and FID 21.8 (50 Hours, 1000 steps). The results indicate that diffusion-based generation produce fewer artifacts with stable training but require higher computational resources and longer training duration.

Attribute	GANs	Stable Diffusion
Strengths	Fast generation, well-established framework	High image quality, stability, diversity
Weakness	Mode collapse, training instability	High computational cost, slower generation
Scalability	Highly scalable with larger models	Scalable but with increased computational demands
Flexibility	High, adaptable to various tasks	Moderate, focused on high-quality synthesis
Training Complexity	Moderate	High
Potential Applications	Real-time image generation, artistic applications	Medical imaging, high-quality content creation

Table 1: Strengths and Weaknesses of GANs and Stable Diffusion [23]

Table 1 presents a comparison of the strengths and weaknesses of GANs and Stable Diffusion models across key factors such as scalability, flexibility, training complexity, and potential use cases. It highlights that GANs are more scalable and faster, making them suitable for real-time applications. Whereas, Stable Diffusion models are better suited for generating high-quality, diverse images, but require more computational power.

In 2025, Yang and Wang published a comprehensive survey in the "International Journal of Computer Vision" for the synthesis of images using generative models when only limited training data are available [24]. The GAN architecture-variant models such as PG-GAN, BigGAN, and the StyleGAN variants can synthesize realistic images when trained on large-scale datasets, but when the training data is limited, problems such as overfitting, memorization, and non-convergent training may arise. VAEs are generally easier to train compared to GANs, but the images they generate often appear blurry or noisy. Optimization becomes difficult for VAEs in few-shot scenarios, which leads to unstable training and unsatisfactory synthesis performance. The authors classify limited data synthesis tasks into four categories: **data-efficient generative modeling, few-shot generative adaptation, few-shot image generation, and one-shot image generation**. They provide overview of state-of-the-art generative models in few-shot settings and highlight issues that arise when these models are trained on limited data samples.

To improve the synthesis quality of **data-efficient generative models**, the authors classify solutions into augmentation-based, regularization-based, architecture-variant-based, and off-the-shelf-model-based approaches. The augmentation-based approaches increase the training sets through data-augmentation. The regularization-based techniques stabilize the model training by imposing additional constraints on the training process. The architecture-variant-based techniques improve data efficiency of generative models by designing suitable network architectures, which includes lightweight GANs like FastGAN to reduce computational cost. The off-the-shelf model-based approach introduces pre-trained models for improving image synthesis under data constraints. The introduction of pre-trained models into GAN training can significantly improve sample efficiency and convergence speed, while reducing training time, since only a small number of parameters need to be trained from scratch. The Faster ProjectedGAN was developed to accelerate training time and reduce computing resources. The benchmarks for evaluating data-efficient generative models consist of subsets of large-scale datasets (e.g., FFHQ and AFHQ) and single-category low-shot datasets. The performance of the models was quantitatively evaluated using FID scores. The experimental results showed that the best-performing model on the FFHQ dataset was FakeCLR + ADA (a regularization-based approach), which

achieved an FID score of 7.25 when trained on 5K samples. On the AFHQ dataset, ProjectedGAN (an off-the-shelf model-based approach) achieved the lowest FID score of 2.16 on the AFHQ-Cat subset using approximately 5K training samples. For low-shot datasets, CHAIN (an architecture-variant-based approach) demonstrated the best performance, obtaining an FID score of 6.93 when trained on the 100-shot Panda dataset. The data-efficient generative model approaches indicate strong data efficiency when trained from scratch on limited datasets, but there are limitations, they often tend to overfit and memorize the training samples, which leads to poor generalization and degraded synthesis, producing repetitive outputs with reduced diversity. The architecture is often computationally expensive, which limits the real-world use.

The **few-shot image generation** aims to synthesize diverse and photorealistic images using few input images from that category as conditional input. The existing methods are commonly grouped into optimization-based, fusion-based, transformation-based, and inversion-based approaches. The transformation-based approaches capture inter-category and intra-category translation to produce new images. The DeltaGAN is an example that improves output diversity. The fusion-based approaches generate novel samples by interpolating the conditional input images at the image or semantic level. The F2GAN suffers from fine-grained details. The local fusion GAN model (LoFGAN) was proposed to address the issue, which introduces local semantic fusion, resulting in high fidelity and improved synthesis quality. The WaveGAN was proposed to improve the fidelity of the generated images. The diffusion-based approaches are also included in few-shot image generation, the few-Shot Diffusion Models (FSDM) capture image distributions at both the image and patch levels and generate novel samples from only a few unseen images at test time. To increase diversity in the images generated, the Conditional Distribution Modeling (CDM) has been proposed to better approximate the conditional distributions of novel categories under limited reference data. The three popular benchmarks were used to evaluate few-shot image generation, which includes Flowers, Animal Faces, and VGGFace datasets. The performance of the models was evaluated using FID and LPIPS scores. In the 3-shot setting, LSO (an inversion-based approach) achieved the best synthesis fidelity (lowest FID) across all three benchmarks, with FID scores of 34.59 on Flowers, 23.67 on Animal Faces, and 3.98 on VGGFace. In terms of diversity (LPIPS), different methods

performed best on different datasets in the 3-shot setting, the DiscoFUNIT (a transformation-based approach) achieved the best LPIPS score on Flowers (0.51), while SAGAN (a fusion-based approach) achieved the best LPIPS scores on both Animal Faces (0.57) and VGGFace (0.42). HAE (an inversion-based approach) achieved the best LPIPS on VGGFace under the 1-shot setting (LPIPS = 0.59). It is observed that transformation-based approaches can produce more diverse images (higher LPIPS), but they may struggle when inter- and intra-category translation is complex, which can limit synthesis quality and diversity. The fusion-based approaches achieve better FID scores, but they may suffer due to poor semantic fine-grained details, and the Inversion-based approaches might not fully exploit the continuity of the latent space, leading to suboptimal generation results.

The **one-shot image generation** synthesizes novel and diverse images by training a generative model using only a single image. The models are expected to capture the internal distribution of the training image. The authors classified one-shot generative models into two categories: parametric and non-parametric approaches. The parametric approaches have produced the best results overall, due to their ability to improve synthesis quality, training speed and enabling fewer parameters (e.g., ConSinGAN), generate novel images, and depending on the method preserve the overall structure and appearance of the input image. The parametric approaches are categorized into GAN-based and diffusion-based models depending on which generative model is used for training. In GAN-based approaches, SinGAN can capture the internal distribution of patches within the input image and generate novel image. ConSinGAN can improve the training speed and achieve better synthesis quality than SinGAN. ExSinGAN provides better generation quality and supports manipulation of the input image. In diffusion-based approaches, SinDiffusion can progressively generate images over timesteps and produce diverse outputs. SinDDM can produce novel samples. and SinFusion can better preserve the overall structure and appearance of the input image. The non-parametric are patch-based approaches which require no costly training and can generate images with better visual quality with fewer artifacts. GPNN is a patch-based algorithm, which uses nearest neighbors to replace similar patches with their closest matches and then combines multiple patches to form a novel image. The one-shot image generation quantitative results showed that parametric methods tend to produce more diverse samples, where SinDDM achieved the highest

diversity with LPIPS Div. (0.21 ± 0.08), SinGAN achieved LPIPS Div. (0.18 ± 0.07) and ConSinGAN (0.15 ± 0.07). In terms of fidelity measured by SIFID, ConSinGAN (0.09 ± 0.05) and SinGAN (0.15 ± 0.05), while SinDDM has a higher SIFID (0.34 ± 0.30). The non-parametric approach GPNN achieved LPIPS Div. (0.10 ± 0.07) and SIFID (0.05 ± 0.04) which showed that the patch-based models achieve better patch-level distribution matching, with higher fidelity. The non-parametric models suffer from low diversity because the synthesized images can be viewed as rearrangements of the internal patches of the referenced image.

In 2025, Wang et al. published a paper at the "IEEE Transactions on Pattern Analysis and Machine Intelligence" on SinDiffusion, which is based on DDPMs to learn internal distribution of patches from a single natural image for diverse image generation with similar visual content [25]. SinDiffusion learns the internal patch distribution rather than memorizing the entire image. A SinDiffusion model trained on a single image can produce high-quality and diverse images of arbitrary resolutions. SinDiffusion uses a single scale training scheme with a single patch-wise denoising network with a patch-level receptive field to capture the internal patch distribution. The receptive field covers the whole training image in a vanilla diffusion model, which often reproduces the training image with limited variation. For photorealistic and diverse images, SinDiffusion predicts the noise with a patch-level effective receptive field, so each prediction depends only on local context. This allows the model to capture localized patterns and textures from the single image's internal patch distribution.

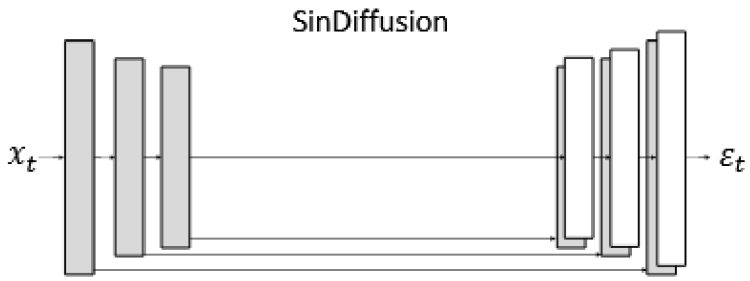


Figure 7: SinDiffusion Denoising Network based on [25]

Figure 7 illustrates the SinDiffusion patch-wise denoising network, which is used in the reverse diffusion process to predict the noise component at each timestep t. The network is a U-Net based, which consists of an encoder and a decoder that takes noisy input image $x_t$and estimate the corresponding noise ε_t. The encoder and decoder each have three stages whose feature-map resolutions are 1, ½ and ¼ of the input resolution, and the channel sizes in each layer set to 64,128,256 respectively. Each encoder layer contains one time-embedded Resblocks and each decoder layer contains two time-embedded Resblocks. This enables timestep-conditioned denoising [26].

The authors designed four network structures with different receptive fields and trained them on a single natural image. It was observed that with a smaller receptive field SinDiffusion tends to generate more diverse images. However, a model with an extremely small receptive field could not preserve the reasonable structure of the overall image. Therefore, a suitable receptive field is required to capture meaningful patch distributions. The depth of the denoising network plays a key role in determining the receptive field size, reducing the downsample and upsampling stages decreases the receptive field. As depth is reduced, the attention layers used in deeper stages are naturally removed, which yields a fully convolutional network. This design choice underlies SinDiffusion's ability to generate images at arbitrary resolutions. The receptive field can also be further constrained by reducing the number of the time-embedded resblocks at each resolution. Hence, by redesigning the model, a patch-wise denoising network with a proper receptive field can be obtained, which results in generating photorealistic and diverse images. The SinDiffusion model is trained using the standard diffusion objective, which is predicting the involved noise conditioned on the timestep. Once trained, the model generates diverse images through an iterative denoising process. The model was trained on each image in Place50 image dataset, which contained 50 natural landscape images and evaluated the fidelity and diversity of the results. To assess visual fidelity, SIFID was used, and generation diversity was evaluated using LPIPS. The SinDiffusion model was trained using AdamW optimizer, and the total timestep was set to 1000, adopting exponential moving average (EMA) with 0.9999 decay. The framework was implemented using Pytorch and the experiments are performed on NVIDIA Tesla V100. The qualitative evaluation shows that SinDiffusion generates diverse and realistic images at different resolutions and aspect ratios that

preserve similar local patch distribution to the training image. With the enhanced SinDiffusion containing 4 downsample layers and an additional time embedded resblock on each scale, a high resolution long-scroll image can be generated. SinDiffusion achieves state-of-the-art performance on the Places50 dataset against other generative models, with SIFID = 0.06 (lower is better) and LPIPS = 0.387 (higher is better), performing better than the baselines, such as ConSinGAN (0.06/0.305), SinGAN (0.09/0.266), ExSinGAN (0.10/0.248), SinDDM (0.34/0.210), and SinFusion (0.64/0.368). A paired user study on quantitative performance also prefers SinDiffusion to each baseline, 65% vs SinGAN, 77.5% vs ExSinGAN, 81% vs ConSinGAN, 75% vs SinDDM, and 68% vs SinFusion, supporting the quantitative results. GPNN produces realistic images but has limited generalization and its outputs often resemble the training images. PatchGN show unreasonable structure and artifacts in the qualitative results. The comparison with large-scale diffusion models, such as StableDiffusion on diverse image generation resulted in, SinDiffusion generating images with a similar internal patch distribution to the given image, while the StableDiffusion model often introduces irrelevant patches or structures that do not match the given image's internal patch distribution. The SinDiffusion trained model generates realistic and diverse images at the expense of time consumption. The authors adopt DPM-Solver to reduce reduce inference time, for fast sampling, generating images with substantially fewer inference steps while maintaining high fidelity.

3.2 IR-Camera Data Synthesis

In 2025, Suárez and Sappa published a paper titled "Thermal Image Synthesis: Bridging the Gap Between Visible and Infrared Spectrum" for generating high quality synthetic thermal images from visible images [27]. The approach addresses the critical challenge in thermal imaging, such as the cost and lower resolution thermal sensors. The authors propose a GAN based framework for generating synthetic thermal images from visible images without the requirement of paired data, which has numerous practical applications such as surveillance, medical imaging, search and rescue operations. The proposed architecture is based on CycleGAN that allows unpaired domain translation and no need of paired visible-thermal datasets. This work incorporates several enhancements to the traditional GAN framework

such as contrastive loss, a relativistic loss, identity loss and spectral normalization. The contrastive loss is implemented to improve the quality of the images obtained. The relativistic loss considers that within each mini-batch, at least half of the samples generated are fake. Based on this assumption the learning divergence is minimized, leading to training stability, and the model can better distinguish between real and synthetic samples. As a result, stable training process, improved image quality and more accurate synthetic thermal images. Identity loss is employed to preserve thermal intensity and object shape during unpaired translation. Finally, the spectral normalization is applied to enhance the quality of the synthetic thermal images generated by the model. Choosing the right color space is crucial for synthesizing thermal images from visible spectrum images. The preprocessing step involves converting visible images into the HSV (Hue, Saturation and Value) color space, and only the brightness (Value) channel is considered for model training, this significantly improves the model's ability to simulate object temperatures accurately. Training is performed using the M3FD dataset, consisting of 4,500 paired images of visible thermal pairs, and for testing FLIR ADAS V2 and a proprietary Thermal Stereo dataset is considered. All images are resized to 256×256 pixels during the training process, utilizing the Adam optimizer with a learning rate of 0.000273. The training process took over approximately 96 hours utilizing TITAN V GPU. To evaluate the quality of synthesized thermal images, structural similarity index (SSIM) and the peak signal-to-noise ratio (PSNR) were used. The proposed method performed better in comparison with several other benchmarks such as original CycleGAN, CUT and FASTCUT models on the training as well as the testing datasets. The experimental results demonstrate the effectiveness of the proposed framework in synthesizing high-quality thermal image.

In 2023, Khan proposed an unpaired image translation framework based on CycleGANs for translating RGB images into corresponding thermal images, mainly focused on applications in low-light visual tasks such as pedestrian detection and security surveillance, where the visible cameras lack the capability to capture high-resolution images at night [28]. Although thermal cameras can provide heat variations of objects through thermal images, their cost is high compared to visible cameras. The proposed methodology uses a two-stream network, comprising of a generator and discriminator models trained from scratch. The

first generator maps RGB images into thermal images with its corresponding discriminator, while the second generator reconstructs RGB images from thermal images along with its discriminator, ensuring cycle-consistency. The framework optimizes both adversarial losses to ensure that the generated thermal images are indistinguishable from real thermal images and optimizes the cycle-consistency loss as well. The generator architecture is taken from Johnson et al., where each CycleGAN generator consists of an encoder, transformer, and decoder. whereas the discriminator network uses a 70×70 PatchGANs, to evaluate the image patches, which helps in improving the capability of image generation. The RGB and thermal image were collected using a Parrot ANAFI Thermal drone, during midday, evening, and night from two different places at UMBC. The dataset consists of 500 images of RGB and thermal set, where the size of RGB image is 1920×1080 pixel, while the size of thermal image is 1440×1080 pixel. The preprocessing includes resizing, orienting, color corrections, augmentation such as flipping and random rotations to increase the samples. All the images were normalized and resized to 256×256 resolution for training. For model training, least-squares loss was applied for stability, with an Adam optimizer having learning rate of 0.001. The batch size was set to 1 for leveraging instance normalization rather than batch normalization. The network was trained for 100 epochs using 400 images for training and 70 for testing. The qualitative results demonstrated that the translating images are little bit similar to original RGB images indicating shortage of data, whereas quantitative evaluation using Mean Squared Error (MSE) confirmed the similarity between generated and real thermal images, which validates the possibility of using the GAN based approach.

In 2021, Han et al. published a paper at the “IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops”, proposing a novel method called Dual Contrastive Learning GAN (DCLGAN) [29]. The study addresses the challenges in unsupervised image-to-image (I2I) translation by learning mappings between domains without paired training data. The method is based on the Contrastive Unpaired Translation (CUT) framework, for efficient mapping between unpaired data, that significantly improves upon existing methods by overcoming limitations related cycle-consistency assumptions, and mode collapse. Approaches such as CycleGAN, DiscoGAN, and DualGAN use the assumption of cycle-consistency for a bijection mapping between two domains. The cycle-consistency

assumption ensures that the translated images have similar texture information to the target domain but fail to perform geometric changes. CUT introduced a patch-based, multi-layer PatchNCE loss to maximize mutual information between corresponding patches of input and output images. However, CUT has drawbacks due to having a single encoder for both domains, which may not efficiently capture the domain gap, and they suffer from mode collapse. To address these issues, the authors proposed DCLGAN, which employs different encoders and dual projection heads for different domains, thereby learning suitable embeddings. The dual learning setup enables mutual information maximization in both translation directions (X→Y and Y→X) and helps to stabilize training process. The framework combines 3 losses including adversarial loss, PatchNCE loss, and identity loss. A variant of DCLGAN is SimDCL which has additional similarity loss to address mode collapse. The experimental evaluation of the DCLGAN, SimDCL and the baselines were conducted on six different datasets on nine tasks, including benchmarks like Horse <-> Zebra, Cat <-> Dog, CityScapes, Van Gogh <-> Photo, Label <-> Facade, and Orange <-> Apple. The performance was measured using FID score and the results demonstrated that DCLGAN and SimDCL outperform baselines such as CycleGAN, CUT, MUNIT, DRIT, and GcGAN. Particularly, DCLGAN achieved state-of-the-art performance in both texture and geometry changes with negligible artifacts. The Cat <-> Dog translation illustrated the model's ability to capture structural changes where other methods failed to generate realistic images. On comparison with the supervised methods like Pix2Pix, CRN and DRPAN on CityScapes dataset demonstrated DCLGAN performing on par with supervised methods in pixAcc significantly closing the gap between unsupervised and supervised learning methods. The drawback of the methods based on mutual information maximization (CUT and DCLGAN) is they cannot prevent mode collapse in Photo -> Label and similar tasks. To address this, SimDCL was designed, and it was tested on Facade -> Label task. From the visual results it was observed that SimDCL generates reasonable output for different inputs compared to other methods including DCLGAN, CUT, CycleGAN.

In 2025, Ma et al. published a paper in the "Springer Nature journal Complex & Intelligent Systems" proposing IRKDGAN [30]. It's a discriminator guided, visible to infrared image

translation technique based on pix2pix framework. The proposed method uses a generator to convert a visible image into the corresponding infrared image, while the discriminator judges the authenticity of the generated infrared image, and the infrared image is reconstructed using the features extracted from the discriminator, distilling the knowledge to the generator's decoder The generator uses a ConvNeXt encoder and a U-Net–style multi-scale fusion decoder (MSF) that fuses encoder features effectively during upsampling, while a standard PatchGAN serves as the discriminator. During inference, only the generator is used. The experiments were conducted using an Intel Core i9-10980XE CPU running at 3 GHz on an Nvidia RTX 3090 GPU, on IRKDGAN and other image translation methods across datasets KAIST, FLIR, and AVIID-3, containing visible-infrared pairs that are aligned. The quality of the generated images was evaluated using six metrics PSNR, SSIM, MS-SSIM (higher is better) and LPIPS, FID, KID (lower is better). The paper reports that IRKDGAN is the best method on image-quality metrics, with qualitative figures. In case of Aerial Visible-Infrared Image (AVIID) dataset, AVIID-3 (1,024 train / 256 test) was chosen for experiments containing diverse scenarios with more complex backgrounds. The dataset was used to verify the authenticity of the infrared images generated by IRKDGAN in the field of UAV remote sensing. IRKDGAN performed well on all six metrics when compared against the baselines methods, which includes IR-GAN, Pix2Px, InfraGAN, ThermalGAN and MRFFF.

In 2023, Ma et al. proposed IR-GAN in "The Visual Computer journal", a CGAN framework for visible-to-infrared image translation [31]. The model addresses the common failures occurred while producing IR images using the previous methods such as texture loss and edge distortion by enhancing both generator and discriminator architectures. The generator is a U-Net architecture based on ConvNeXt, which improves texture information, while a Multi-scale PatchGAN is used as a discriminator, to extract multi-scale image features. The gradient vector loss is introduced in the generator to improve the ability of the generator to extract edge information from images. The authors summarize the limitations and contributions of the previous models for infrared image generation, where Pix2Pix provides a supervised image-translation framework but is insufficient for IR image generation tasks, ThermalGAN and LayerGAN have the limitations of high dataset

requirements, I-GANs improve the generator using advanced techniques but neglect discriminator design, InfraGAN introduced U-NetGAN for IR image generation but neglects improving the generator, and DAGAN introduced dual-attention and flashover prediction but lacks objective evaluation of IR image generation.

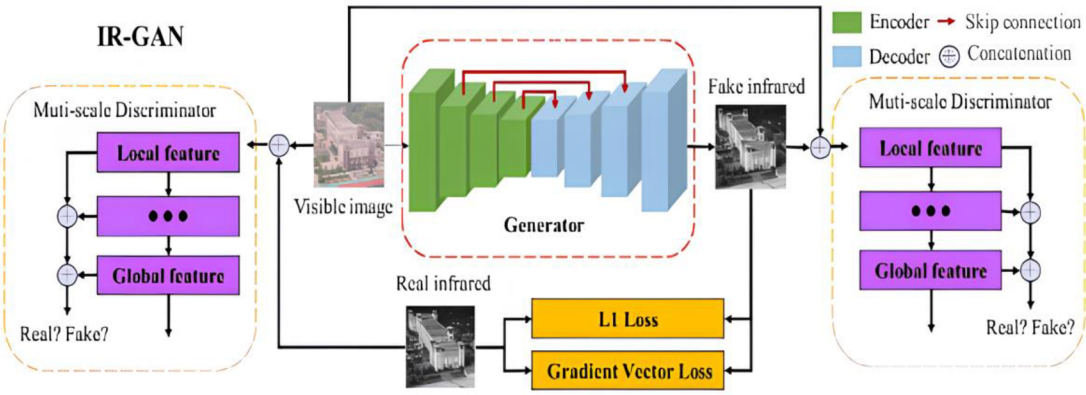


Figure 8: Algorithm Structure of IR-GAN [31]

Fig. 8 illustrates the overall structure of the proposed IR-GAN. A generator with an encoder–decoder architecture translates visible images into fake infrared images, assisted by skip connections that preserve low-level details. The output is optimized using L1 loss for global similarity and gradient vector loss for edge preservation. The multi-scale discriminator evaluates both the real infrared images paired with visible inputs and the fake infrared images generated by the network, enabling it to analyze local and global features, distinguish authentic samples from synthetic ones, and provide feedback to improve the generator. Through this adversarial training process, the generator progressively learns to produce infrared images. IR-GAN and other GAN-based methods including Pix2Pix and ThermalGAN were trained and evaluated using VEDAI and IVFG dataset, consisting of IR and visible image pairs. The VEDAI dataset consists of aerial vehicle imagery with 1,046 pairs for training and 200 for testing cropped to 256×256 for network input. The IVFG dataset contains five object categories (buildings, radars, generators, cooling towers, ships) with 1,544 images used for training and 431 images used for testing, corresponding to an approximate 7:2 split. IR-GAN was implemented on the improvements

of Pix2Pix framework and trained for 200 epochs. The Adam optimizer was employed with the parameters were set as $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The initial learning rate was fixed at 0.0002 for the first 100 epochs, after which it was linearly decreased to zero over the remaining epochs to ensure convergence. The loss function incorporated the adversarial loss, with the L1 reconstruction loss set to $\lambda = 100$ and the gradient vector loss set to $\varphi = 22$ to improve quantitative performance. The training process was performed on an NVIDIA RTX 3090 GPU. The performance of the models was evaluated using PSNR, RMSE, SSIM, MS-SSIM, and FSIM metrics. IR-GAN achieved the best scores across all metrics on both datasets. From the visualizations, it was observed that the image generated by IR-GAN maintained better structural properties, exhibited clearer contours, and was closer to the real infrared image. Ablation experiments on IVFG dataset validated the effectiveness of UConvNeXt, multi-scale PatchGAN, and gradient vector loss, showing improvements across evaluation metrics.

3.3 RADAR Sensor Data Synthesis

In 2023, Fidelis et al. introduced a method for synthetic raw radar data generation using GAN, which generates intermediate Frequency (IF) signals from FMCW radar [32]. The authors employed WaveGAN-based architecture, specifically WGAN-GP. The dataset containing radar measurements were one-dimensional with multiple channels, so the architecture used was one-dimensional convolutions. The method was trained using radar measurements collected from a motorcycle traveling in a straight line towards the radar. It receives two inputs, the distance of the motorcycle from the radar and Gaussian noise. These inputs are processed by the generator to synthesize 16 simultaneous radar chirps, mimicking the original IF signal from FMCW radar systems. The trained model tries to generate signal measurements that have similar characteristics to that of real measurements obtained in the real world. The authors used FID metric to evaluate the synthesized radar data. The FID score of the training data, test data and generated synthetic data were calculated. The FID scores closer to zero means that the generated synthetic data are better. It was observed that the FID score between the generation and test datasets (18.12) was a bit lower than the score between the training and test datasets (18.18), which

indicated that the synthetic data more closely resembled the real test data. Similarly, the FID score between the generation and training datasets was 0.51, which indicated that the synthetic data resembled closely with the training data. The Euclidean distance, which is a Nearest Neighbor (NN) technique, is the other metric used to check if the generated data is new and not copied from the training data. The NN value for generation and training (2.37) is higher than training and training (1.00) which showed that the synthetic data samples were not identical to the training set. The authors also evaluated the synthetic radar data by generating Range-Azimuth (RA) maps from both real and synthesized radar signals. Based on these RA maps an algorithm was used for object detection. The results confirmed that the generated synthetic data is realistic in terms of coherent radar reflections of the motorcycle and background noise based on comparison of chirps, RA maps and object detection results. A regression model was used to estimate the motorcycle distance from the radar using the generated data. The model achieved a mean absolute error of just 2.69 meters, having a range of possible values between 0 and 25 meters. This demonstrates the applicability of the synthesized radar signals.

In 2020, Oliveira and Bekooij presented their work at the "IEEE Radar Conference" [33] on radar data synthesis by proposing the combination of GANs and Deep Convolutional Autoencoders to generate synthetic Range-Doppler (RD) maps of FMCW radars, which involves moving objects like pedestrians and cyclists. The authors have used Artificial Neural Network (ANN) model for generating new data. The proposed methodology is based on Generator-Predictor system. The GAN is used as generator to generate synthetic RD map for input containing black boxes that indicates the origin positions of moving objects, whereas a Deep Convolutional Autoencoder (CAE) network is used as a predictor to generate the next four future frames for the RD map created by the generator. The synthetic RD maps were generated using real RD maps obtained from the FMCW radar. The 77 GHz TEF810X FMCW radar was used to obtain raw data. Two-dimensional FFT was applied on raw data to obtain RD maps. The obtained dataset consisted of 2495 RD maps, from which 1898 RD maps were used for training, while 547 RD maps were used for final evaluation, and the remaining 50 RD maps which contained no moving objects were used to produce noise. Each RD map has a size of 128×128 matrix, which were converted into black and

white images and were labeled manually. The training frames were denoised using three Convolutional Autoencoders (CAEs) in parallel, which process sequence of three frames to produce a single clean frame with only the moving objects. This system reconstructs with an accuracy of 96.4% for objects in the denoised RD maps. To obtain a larger training dataset, the data augmentation technique such as mirroring was applied on the frames, which gave a total of 4008 frames, which were then used to train both generator and predictor. The Generator-Predictor system is used to create synthetic RD maps. The three types of networks in this framework are the GAN based generator and discriminator, and CAE based predictor. The generator accepts input containing black boxes, and the discriminator is fed with actual RD maps for comparison. During training, the generator produces a synthetic RD map with moving objects, while the discriminator distinguishes between real and synthetic data. Next, the predictor takes the synthetic RD map from the generator as the input and outputs five synthetic frames of moving objects, containing current RD map plus four future predicted RD maps. The visual comparison shows that the predicted frames resemble actual frames. This noise is added to the generated sequential frames (ground-truth) from the predictor to obtain noisy output. This noisy output mimics the real RD maps that were obtained after processing actual radar data. To evaluate the synthetic RD maps, the authors built a detection system that consists of a new CAE and a foreground detector. This system was trained using only synthetic RD maps. It achieved 95.2% accuracy for detecting objects when tested on real RD maps. For performance comparison, the authors trained the system using different amounts of real radar data and observed that the synthetic data showed better performance over small real datasets and sufficiently close performance over fully collected FMCW radar dataset. Another method of evaluation was performed through PSNR metric, which is used for image quality evaluation. A higher value indicates that the retrieved image is close to the original, while a lower value means a high amount of noise. The mean PSNR for the detection system using only synthetic data is 28.35 dB, while the mean PSNR for the detection system trained with 4008 groups of real data is 28.56 dB. The noise difference is close, which shows that the generated synthetic data is a good generalization of the real data.

In 2023, Wachtel et al. presented a paper at the "IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)" for the synthesis of FMCW radar RD targets with Micro-Doppler using Conditional GAN (CGAN) [10]. It addresses the challenge of acquiring data using high-cost equipment by proposing GAN-based method to achieve synthetic data like real ones. The research builds upon the previous work by De Oliveira and Bekooji on the RD image generation, while improving the architecture, data collection, generated data, and analysis of the results. The authors used INRAS RadarLog FMCW radar sensor to obtain the data and retrieve raw data, which was processed by applying post-processing algorithm to obtain RD images. The algorithm calculates the image by Doppler shift effect principle. This effect is caused by the motion between the object and the radar, where the frequency difference between transmitted and received signals carries the range and velocity information, which can be obtained by three Fast Fourier Transforms (FFTs). The resulting dataset consists of 13,561 cropped images of RD targets (with Micro-Doppler), each having a size of 64 × 64, and sorted into four categories: Pedestrian, Bicycle, E-Scooter, and Noise. The architecture developed by the authors is a Conditional GAN with Deep Convolutions, where the networks were adapted to fit the radar images by utilizing only the single-channel intensity points rather than three-channel RGB format.

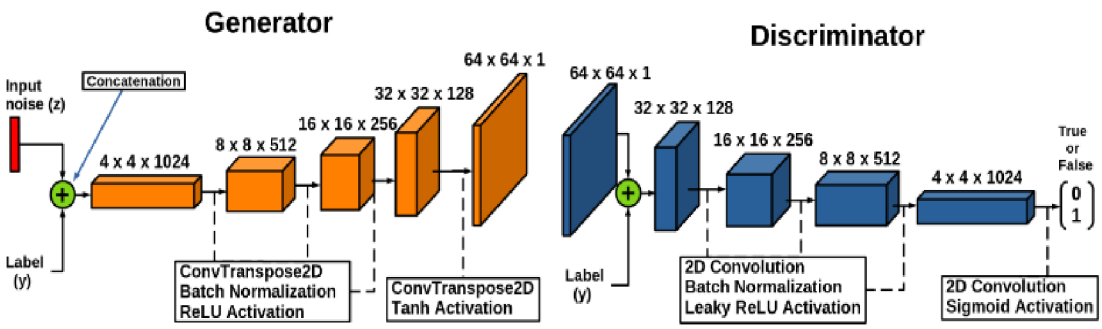


Figure 9: The CGAN architecture used [10]

Figure 9 illustrates the authors proposed Conditional GAN with Deep Convolutions architecture, where the networks were adapted to fit the RD images. Instead of using three channels for RGB colours, only a single channel (64 × 64 × 1) was used because the image

consists of intensity points with no colour information. The same class label was concatenated with the input noise vector for the generator and with the input image for the discriminator, allowing for the generator to create target specific images from random noise, and the discriminator to distinguish between real and synthetic images. The parameters used in the training process were Adam Optimizer, learning rate of 0.0001, and 200 epochs. The evaluation of the synthesized images was performed by testing a pre-trained CNN classifier and measuring the data augmentation capacity. The pre-trained CNN classifier achieved 95.9% accuracy on the real test set, and 82% accuracy on the generated test set. The results indicated that the model generated realistic images, which were close to real ones. Also, the 4,500 generated samples were used for data augmentation with the real training samples, resulting in 18,061 training samples. The CNN was retrained with the augmented data, which achieved an overall accuracy of 96.5% on the real test dataset. The confusion matrix indicated that, the data augmentation led to a slight decrease in accuracy of pedestrians and bicycles, and an increase in accuracy for E-Scooter. The E-Scooter caused the 0.6% increase in overall accuracy, which is not significant and remains almost the same. This indicated that the generated data was neither realistic nor unrealistic.

3.4 Laser Scanner Data Synthesis

In 2025, Nakashima et al. presented a paper at the "IEEE International Conference on Robotics and Automation (ICRA)" introducing R2Flow, a fast and high-fidelity generative model for LiDAR data synthesis [34]. The model was proposed to address the challenge faced by diffusion models, as the diffusion models have proven effective in improving training stability and generation quality, but require numerous iterations of running neural networks to generate high-quality samples leading to high computational costs. R2Flow addresses these challenges by leveraging rectified flows. Rectified flows represent data generation as iterative transformations using deep neural network. They use deterministic straight trajectories, which makes sampling robust to number of steps and enables the entire trajectory to simulate in just one step. R2Flow generates high-quality samples across different number of iterations, whereas LiDM and R2DM suffer from trade-off between quality and sampling efficiency in their iterative generation process. The vanilla VAE, basic

GAN, DUSty, and DUSty v2 methods were used for the synthesis of LiDAR range images. The drawbacks of these methods are, VAEs provide stable training but often produce blurry outputs, whereas GANs achieve better sample quality than VAEs but suffer from unstable training, and their generated point clouds still deviate from real data samples. The score matching with Langevin dynamics (SMLD) framework is used in LIDARGen, and DDPM framework is used in R2DM. Both use pixel-space iteration which can capture fine details, but they require high computational costs due to the iterative nature of sampling. To solve this, LiDM with architectural improvements based on the latent diffusion model (LDM) was proposed. However, LDM-based approaches struggle with blurriness. R2Flow uses pixel-space iteration to ensure pixel-precision for range images. R2Flow learns a transport map between Gaussian distribution and data distribution through data transformation, which is formulated as an ordinary differential equation (ODE). The training procedure involves multiple stages. In the first stage of 1-RF, the model trains to align predicted velocities with uniform interpolation paths. However, these trajectories are not straight and remain curved because the 1-RF is trained on independent data pairs. Hence, 1-RF still requires a high number of sampling steps. To address this, a second stage known as straightening is applied, to address the curve by iteratively refining the flow field through reflow, which leads to the sample trajectories becoming straighter. As a result, the model obtained by reflow is called 2-RF, which produces straighter trajectories. The training is improved by applying a pseudo-Huber loss and a U-shaped timestep sampling strategy. Final stage is timestep distillation, the straightened 2-RF model is further improved by training on selected timesteps (2-RF + k-TD, where k can be 1, 2 or 4) and sacrificing predictions at the other unnecessary timesteps. The rectified flows support inversion. It allows the transformation of data samples into corresponding latent space, which is achieved by solving the ODE in reverse direction from data space back to Gaussian latent space. This shows that, learned rectified flows can transform between different LiDAR scenes. R2Flow is trained on the equirectangular image representation of LiDAR data having measures of range and reflectance at each laser angle. The range modality is rescaled to a log-scale representation to improve geometric resolution for nearby points. The generated range images can be projected back into three-dimensional point clouds with the reflectance values. R2Flow utilizes Hourglass Diffusion Transformer (HDiT), which is a Vision

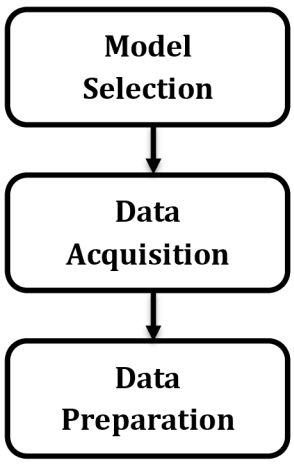
Transformer (ViT) based architecture proposed for pixel-space diffusion models. To adapt HDiT for panoramic LiDAR generation (range and reflectance image) architectural modifications were introduced. The evaluation was conducted on the KITTI-360 dataset by utilizing 81,106 point clouds captured using Velodyne HDL-64E LiDAR sensor. R2Flow was compared against baseline methods, which includes GAN-based methods (Vanilla GAN, DUSty v1, DUSty v2) and diffusion-based methods (LIDARGen, R2DM, LiDM). The evaluation metrics used were Fréchet range distance (FRD), Fréchet range image distance (FRID), Fréchet point cloud distance (FPD), Fréchet point-based volume distance (FPVD), Fréchet sparse volume distance (FSVD), Jensen–Shannon divergence (JSD), and maximum mean discrepancy (MMD), which are based on five types of data representations that includes range images, reflectance images, point clouds, voxels, and bird's-eye views (BEV). The quantitative results indicated that among the GAN-based methods, DUSty v2 performed best across multiple metrics. However, iterative models outperformed GANs. Among the iterative models with a high Number of Function Evaluations (NFE), LiDM with Absolute Positional Embeddings (APE) showed improvements in BEV-based metrics (JSD and MMD). R2Flow and R2DM both achieved comparable results with 256 NFE. While other iterative methods exhibited performance degradation with fewer NFE, R2Flow mitigated the degradation through reflow and distillation (2-TD and 4-TD). This is observed by FRD scores, where R2Flow outperforms baselines such as DUSty v2, R2DM, and LiDARGen as the number of steps decreases. The qualitative results for the LiDAR point clouds generated from DUSty v2, LiDARGen, improved LiDM, R2DM, and R2Flow were compared. The results indicate that R2Flow, R2DM, and LiDM demonstrate better quality, such as sharper scan lines and clearer object boundaries, while some wavy and blurry boundaries were exhibited by LiDM. The comparison of model architectures for the pixel-space velocity estimator indicates that HDiT-based architecture achieves the best performance. The trajectory curvature of learned flows measured over timestep verifies the straightening effect by reflow.

4 Methodology

This chapter describes the methodology for the generation and evaluation of synthetic images derived from data collected using drone-mounted sensors. The conceptual framework provides a workflow that contains the distinct stages involved in the generation process. The drone is equipped with multiple sensors, where each sensor captures data in a different format. For the generation of synthetic images for each sensor modality, specialized generative frameworks are applied to ensure a high-quality synthesis of images. The following section contains the specific implementation of this pipeline for each sensor type.

4.1 Conceptual Framework

The conceptual framework provides a structured methodology for generating synthetic data from sensor captured real-world data. The different generative models are selected and trained for each type of sensor data to ensure high quality synthetic data generation. This approach of using different sensors and specialized models helps to generate synthetic datasets that are both realistic and diverse, which makes them more useful and reliable for real-world applications. The framework is presented across all modalities, but the experimental evaluation is conducted only on RGB camera data.



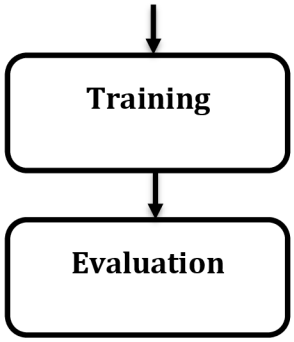


Figure 10: Synthetic Data Generation Process for Sensor data

Figure 10 shows the complete workflow for generating synthetic images using real sensor data. The process begins with model selection. Next is data acquisition. The collected data then goes through dataset preparation. After this, training the generative model using the prepared data to generate synthetic samples. Finally, data evaluation is carried out.

Model Selection:

It focuses on selecting the best generative model for the high-quality synthetic image generation across different sensor data based on their demonstrated performance in prior literature review, and their ability to preserve data specific features, and their scalability to the collected sensor data. The objective is to choose a model that generates visually realistic and diverse images. After an extensive review and comparison of state-of-the-art models, the optimal model for each sensor type will be identified. The models are chosen based on the analysis that they are best aligned with the challenges and unique characteristics of each sensor data.

Data Acquisition:

The drone collects data using several different sensors, which include an RGB, an IR camera, a radar sensor, and a laser scanner. The sensor captures data in their own format, which gives a unique view of the environment. To build a dataset, the drone follows carefully planned flight paths at different heights and in different conditions. This helps capture a wide variety of real-world scenarios.

Dataset Preparation:

The data collected from the sensors are in raw formats, which need to be transformed into a model specific representation before training. Each sensor data has its own data structure and encoding. Therefore, preprocessing is performed independently for each sensor data to ensure compatibility with the chosen generative model.

Training:

The prepared data for each sensor type is used to train the selected generative model. Training involves choosing the best parameters to minimize the loss function that ensures that the output closely resembles real data. The hyperparameters are tuned to achieve high fidelity and diversity in the generated samples and prevent overfitting. Regular checkpoints are created to store intermediate model states, which enables the selection based on performance.

Evaluation:

The evaluation of synthetic sensor data is performed using quantitative metrics and qualitative inspection to measure how closely the patterns in the synthetic data match real data and how realistic they appear. The quantitative evaluation includes specific metrics for each sensor output. The qualitative evaluation involves visual inspection to identify artifacts, geometric inconsistencies, or deviations from expected structural patterns. These evaluations ensure that synthetic datasets are statistically aligned with real data distributions while retaining fidelity and diversity.

4.2 RGB Camera Sensor Data

This section explains the methodology for the generation of synthetic images using the real images captured from RGB camera sensors mounted on the drone. It covers the selection of the methods and choosing the best generative method which can produce high-fidelity and diverse images for the available training data. Also, it details the preprocessing steps that

are required for RGB images before training the generative method. Finally, it outlines the quantitative and qualitative metrics used to evaluate the generated images.

4.2.1 Model Selection

The GANs such as DCGAN, WGAN, CycleGAN, StyleGAN, and BigGAN can suffer from training instability and mode collapse. The Diffusion-based models such as DDPM, Improved DDPM, Vanilla Diffusion, and Stable Diffusion generate images with artifacts, and their iterative denoising process requires longer training duration and higher computational resources to train on large-scale datasets such as **ImageNet** [35] (object categories), **CIFAR-10** [36] (natural images), and **CelebA** [37] (human faces) [23]. The generative models designed for the synthesis of RGB images are typically trained on large-scale datasets such as **FFHQ** [38] (human faces), **AFHQ** [39] (cat/dog/wild), and **LSUN** [40] (outdoor/indoor) [24], but collecting large number of data samples from RGB cameras mounted on drones is a challenge because of environmental constraints, sensor cost, flight limitations, and inconsistent data acquisition conditions and also the data collected using the cameras contain varied scenes and they don't contain similar patterns observed in the above mentioned datasets. This research has only a limited number of training samples available, which can lead to a high chance of model overfitting and memorization, and can negatively affect the fidelity (realness) and diversity (variation) of the generated images. Therefore, the right generative model for the RGB image synthesis will be selected, which can adapt to low-data scenarios with strong generalization capabilities.

The survey by Yang and Wang is specifically relevant to RGB camera sensor images, as evidenced by benchmark datasets, and it helps to identify the most suitable generative model under limited data availability [24]. The generative models show excellent performance when trained on large number of training samples but tend to suffer from performance deterioration due to model overfitting, and memorization. The VAEs synthesized images are often blurry and noisy images due noise distribution injection, and the limited data may make them difficult to optimize, which can lead to unsatisfactory synthesis performance and unstable training. Therefore, VAEs are not the best options for

high-fidelity image synthesis from sensor images when only a few samples are available. In the data-efficient generative modeling category [24], several promising approaches were evaluated. There are augmentation-based approaches (which expand the training images through data augmentation), regularization-based approaches (which apply additional constraints to stabilize training), architecture-variant-based approaches (which includes lightweight GANs such as FastGAN to reduce computational cost), and off-the-shelf model-based approaches (which gives pre-trained models to improve sample efficiency and convergence speed). The Faster ProjectedGAN was developed to accelerate training time and reduce computing resources by leveraging the off-the-shelf model-based direction. The GAN models FakeCLR + ADA (Table 2 in [24]), ProjectedGAN (Table 3 in [24]), and CHAIN (Table 4 in [24]) achieved the lowest FID scores, which made them strong candidates for synthetic image generation. However, despite their performance, the data-efficient generative models have limitations, where they suffer from memorization, reduced diversity, often replicating training images, demonstrate poor generalization, and degrading synthesis quality. The architecture designs are computationally expensive, which limits their applicability in real-world scenarios. The benchmarks such as **FFHQ** [38], **AFHQ** [39], and single-category low-shot datasets used for evaluation lack diverse scenarios, and the evaluation metric FID does not capture diversity in the synthesized images. As a result, these methods are not considered the best choices for generating synthetic images for RGB images. In the few-shot image generation category [24], the LSO method achieved the best performance compared to other methods on benchmarks (**Flowers** [41], **Animal Faces** [42], **VGGFace** [43]), achieving the best synthesis fidelity and high diversity as determined by their FID and LPIPS scores (Table 9 in [24]). However, LSO belongs to inversion-based approaches, and these approaches are unable to fully exploit the continuity of the latent space, which can lead to suboptimal generation results and lower-quality images. Other approaches like transformation-based methods may struggle to generate novel images, and fusion-based approaches suffer from poor semantic fine-grained details. Therefore, few-shot image generation methods cannot be considered as the best solution for synthetic RGB image generation.

The parametric one-shot image generation [24] approaches are the best for the images captured using drone-mounted RGB cameras because these approaches provide highly effective solutions for scenarios where collecting large-scale data is not feasible. The generative models are trained on only a single image to capture the internal distribution of the training image and generate novel, diverse images that share the same distribution as the reference image. They achieve best results by improving synthesis quality, increasing training speed, reducing parameter requirements, and preserving the overall structure and appearance of the input image. The parametric approaches are categorized into GAN-based (SinGAN, ConSinGAN, ExSinGAN) and diffusion-based (SinDiffusion, SinFusion, SinDDM) methods, both demonstrating strong performance for RGB image synthesis. The non-parametric one-shot image generation approach GPNN suffers from low diversity, and it has limited generalization and its synthesized images often resemble the training image. The best generative model for the RGB image synthesis is one among the parametric approaches.

Wang et al. proposed SinDiffusion model show that it performed better than other diffusion-based methods such as SinFusion and SinDDM, as well as GAN-based methods such as SinGAN, ConSinGAN, and ExSinGAN, across both quantitative and qualitative benchmarks on the **Places50** [44] dataset [25].

Rank	Method
1	SinDiffusion
2	ConSinGAN
3	SinGAN
4	ExSinGAN
5	SinDDM
6	SinFusion

Table 2: Ranking of the methods for RGB image Synthesis

Table 2 illustrates the ranking of the methods in contention for producing the best, high-quality and diverse synthetic RGB images. The ranking is completely based on the quantitative performance on the **Places50**[44] dataset (Table I in [25]), with higher priority given to fidelity (SIFID) for quality, and then the diversity (LPIPS). The PatchGenCN showed poor qualitative results among the above considered generative methods and the GPNN synthesized images resemble the training image due to which both were not considered for ranking. SinDiffusion achieves state-of-the-art performance on Place50 dataset and is the most suitable choice for synthetic RGB image generation. On the **Places50** [44] dataset, it achieves high fidelity and high diversity simultaneously (low SIFID, high LPIPS). The qualitative results (Fig. 8 in [25]) show that it preserves global structure and avoids artifacts common to GAN-based baselines, SinFusion is more diverse but suffers large fidelity drops, and SinDDM trails on both metrics. The paired user study further confirms perceptual superiority, with users preferring SinDiffusion over other methods. Once trained on a single real image, the model produces high-quality, diverse images at arbitrary resolutions, requiring no architecture changes or finetuning per image. SinDiffusion is also much better than large-scale diffusion models such as Stable Diffusion, as they generate images with a similar internal patch distribution to the given image, while the Stable Diffusion model produced low-quality images with irrelevant patches. Fast-sampling strategy can reduce the timesteps. Therefore, SinDiffusion is identified as the best model for generating high-quality, and diverse RGB images under limited data conditions.

Method	Summary
ConSinGAN	The ConSinGAN can improve the training speed and better synthesis quality than SinGAN based on SIFID score. It matches SinDiffusion on fidelity, but diversity is less than SinDIffusion. The qualitative results show ConSinGAN generated images exhibit artifacts.
SinGAN	The SinGAN captures the internal distribution of patches within the input image and generate novel images with lower fidelity

	and diversity than ConSinGAN. The qualitative results show SinGAN performs better than ConSinGAN with minor artifacts.
ExSinGAN	The ExSinGAN qualitative results show artifacts, but the model has better generation quality than SinGAN and ConSinGAN based on the visual results. The quantitative results for ExSinGAN indicate that it has lower fidelity and lower diversity than SinGAN and ConSinGAN.
SinFusion	The SinFusion qualitative results show that it preserves the overall structure and appearance of the input image. The model generated images have highest diversity after SinDiffusion, but it has the lowest fidelity based on the quantitative results.
SinDDM	The SinDDM model produces novel samples but trails on both fidelity and diversity quantitatively among all the other models. However, the qualitative results are much better.

Table 3: Summary of other contenders

Table 3 illustrates a summary of other contenders for the synthesis of RGB images. It is based on the performance observed on the **Places50** [44] dataset (Table I and Fig. 8 in [25]). If SinDiffusion does not perform well when trained on RGB images, these other methods can serve as alternative candidates for experimentation. However, this research will be limited to generating images for the top ranked model only.

4.2.2 Data Acquisition and Preparation

The RGB images are collected using DJI drone platform, specifically **DJI Mavic 3M** (Multispectral) **camera** array as sensor payload. It is equipped with movable forward-facing and downward-facing high-resolution RGB camera along with multispectral sensors. In this study, only data collected using RGB camera is used. The data collected using the downward-facing camera consists of ground surfaces containing various objects, while the

forward-facing camera captures views of surrounding landscapes and terrain. The data is collected during multiple drone flights in outdoor conditions. The images are stored as JPEG files. All the images used in the experiments are selected from the acquired dataset without any manual preprocessing involved. These images are used in training and evaluation of SinDiffusion without any preprocessing or transformations required.

4.2.3 Training

SinDiffusion is trained on a single prepared RGB image. The framework employs a patch-wise denoising network with residual blocks, downsampling and upsampling layers to control the receptive field size. The receptive field is carefully chosen to balance diversity and structural fidelity. The network depth is reduced by removing deeper-stage attention layers, which yield a fully convolutional architecture that supports synthesis of arbitrary-resolution images. The training involves iteratively reconstructing the input image through predicting noise at each diffusion timestep. The hyperparameters such as number of diffusion steps (timesteps), and optimizer settings are selected to ensure stable convergence and high-quality reconstruction. The AdamW optimizer is used, and an exponential moving average (EMA) of model weights is maintained to stabilize training. Once training is complete, SinDiffusion can generate diverse synthetic images for the real image [25].

Hyperparameter	Description / Settings
Total Diffusion steps	Number of iterative denoising steps (T=1000)
Optimizer	Ensures stable convergence (AdamW)
EMA Decay Rate	Stabilizes training by smoothing weight updates (0.9999)
Residual Blocks Per Scale	Controls network depth and receptive field size

Down / Upsampling Layers	Influences receptive field size
Receptive Field Size	Smaller fields give higher diversity, while larger fields give better structure (Medium)

Table 4: Tunable Hyperparameters for SinDiffusion Training

Table 4 summarizes the key hyperparameters used for training SinDiffusion on a single image. Each setting is chosen to balance fidelity, diversity, and computational efficiency while ensuring stable convergence. SinDiffusion is trained on each image independently, so no train and test split is required. A single image is sufficient to generate real and diverse synthetic images. The model avoids overfitting by selecting a proper receptive field size, so that the model can learn small internal patch patterns from the image instead of memorizing and copying directly from the whole image.

4.2.4 Evaluation

The synthetic RGB images generated by SinDiffusion are evaluated for both fidelity and diversity to determine how well they replicate the structural and statistical characteristics of the original drone acquired RGB images. The **quantitative performance** is measured using SIFID (Single Image Fréchet Inception Distance) and LPIPS (Learned Perceptual Image Patch Similarity) [25]. The SIFID metric is used to assess visual quality. It measures the difference between the patch-wise features of the generated images and the corresponding input RGB image. Lower SIFID values indicate higher fidelity, which means that the generated images closely preserve the local textures, structures, and visual patterns of the training image. The SIFID is reported as min, max, mean, and standard deviation (std) because the SIFID score can be different for each generated image, and reporting only the mean can hide how much the quality changes from sample to sample. The diversity of the generated images is evaluated by computing the average LPIPS distance between multiple outputs generated from the same RGB input. Higher LPIPS values indicate

greater diversity among the generated images. The LPIPS is reported as min, max, mean, and std because LPIPS values vary across different pairs of generated images, and reporting only the mean can hide cases where some images are very similar (duplicates) or very different, so the LPIPS score is computed over all unique pairs of the generated images as well. The **qualitative evaluation** is done by visually inspecting the generated images based on the results from the quantitative metrics to check if textures, structures, and details look realistic and if multiple outputs show meaningful variation. The outputs are inspected for artifacts, stretching, logical flow of the landscape, and to check if the model generates hybrid objects (e.g., merging of two different surfaces). The generated samples are compared with the training image to ensure the model is generating novel arrangements of patches.

Qualitative limitations and edge cases:
The model might fail to maintain geometric logic (e.g., a road ending abruptly). The generated images can contain stretching in regions where background contains clouds. The model cannot generate objects it has never seen in the training image.

Methodology for testing qualitative limitations:
A small set of image samples are selected from the available real samples from drone captured images and multiple synthetic images are generated for each real image. The generated images are evaluated using qualitative visual inspection to check with the real image for geometric consistency, and any unnatural stretching to identify failure patterns for drone captured images.

4.3 IR Camera Sensor Data

This section details the methodology for the synthesis of thermal images from RGB images, where paired RGB-thermal images captured using drone-mounted sensors serve as training images for the generative model. The following subsections describe the model selection process, the specific dataset requirements, the model training involved, and the evaluation of the generated thermal images.

4.3.1 Model Selection

There are multiple approaches that have been proposed for the synthesis of thermal images from RGB images as a cross-domain translation task. It is critical to select a method that generates highly accurate and realistic images for the drone-mounted sensor captured images. The thermal image captured with infrared thermal sensor encode temperature values at the pixel level. The reviewed generative models focus on grayscale thermal representations, which directly capture temperature information. The supervised learning methods are highly effective at learning pixel-wise correspondences between modalities for small-scale and paired datasets, which reduces artifacts and improve both alignment and thermal consistency. The generative models have the capability to translate RGB images into realistic thermal images, which can be applied for the drone collected paired images as well, since the drone used in this research is equipped with both an RGB camera and an IR thermal sensor, which captures the same scene. Hence, the paired image translation technique seems the most suitable approach. This supervised, pixel-to-pixel mapping between RGB features and thermal images enables the synthesis of high-fidelity thermal images for the RGB images.

The CycleGAN-based framework proposed by Suárez & Sappa rely on unpaired image translation technique designed for scenarios which lacks paired RGB–thermal datasets [27]. The method is not suitable in this research because the drone-mounted sensors capture paired RGB–thermal images, which makes supervised paired image translation approaches more effective. The geometric accuracy and object contours must be preserved for aerial imagery which is only possible through supervised learning for paired images. The training process is also computationally intensive, which requires 96 hours of training on a TITAN V GPU for a small dataset, which raises scalability concerns. The model performs better than unpaired baselines like CycleGAN and CUT (Table I in [27]), but it remains less suitable for producing high-fidelity thermal images.

The khan's dataset is closely aligned with this research, consisting of paired RGB and thermal images captured by a thermal drone [28]. However, the qualitative results indicate that the translated images are a little bit like real RGB images and author concluded that the

shortage of data is the reason. However, the main reason is due to reliance on CycleGAN, which is an unpaired image-to-image translation method which learns bidirectional mappings between two image distributions and does not rely on paired supervision, so due to this reason it cannot ensure accurate pixel-level mapping for the paired data. The evaluation also relied only on the Mean Squared Error (MSE), which is insufficient for assessing structural fidelity, and this is critical for aerial imagery, so CycleGAN approach is not considered as the most suitable approach for the synthesis of thermal images for paired RGB-thermal images.

The DCLGAN [29] demonstrates strong performance among unpaired image translation techniques and competes with supervised methods such as Pix2Pix, CRN, and DRPAN in terms of pixAcc, which shows promise in general image translation. It performs better than CycleGAN on benchmark datasets and it remains a valuable option for cross modality image synthesis experimentations. However, DCLGAN has not been validated for visible-to-thermal translation, and its evaluation datasets focus on domains like animals and cityscapes rather than thermal images, which limits its applicability for this study.

Rank	Method
1	IRKDGAN
2	IR-GAN
3	Pix2Pix
4	InfraGAN

Table 5: Ranking of the methods for Thermal image Synthesis

Table 5 illustrates the ranking of competing methods for thermal image synthesis [30]. The ranking is based on the infrared image generation quality evaluation results on the **AVIID-3** [45] (Table 1 in [30]), **VEDAI** [46] and **IVFG** [47] (Table 2 in [31]) dataset, and all these datasets have paired visible-infrared images, which are captured from UAVs and fits for this research very well.

The IRKDGAN[30] has demonstrated state-of-the-art performance across all six quality evaluation metrics (PSNR, SSIM, MS-SSIM, LPIPS, FID, KID) on **AVIID-3** [45] (Table 1 in [30]), which make it as the best model for high-quality infrared image synthesis and, is well suited to synthesize thermal images for our drone captured RGB-thermal image pairs. However, since the authors have not released training code for training the generative model, IRKDGAN cannot be implemented.

The IR-GAN [31] is an effective approach for RGB-to-thermal image translation. It offers a strong balance between quantitative and qualitative performance. Pix2Pix lacks infrared-specific fidelity, and ThermalGAN requires large number of samples for training, whereas IR-GAN addresses challenges such as texture loss and edge distortion, which results in improved structural fidelity in generated images. The quantitative results indicate that IR-GAN performs better than Pix2Pix and ThermalGAN across the **VEDAI** [46] and **IVFG** [47] datasets, and qualitative visual inspection indicates that it produces infrared images that are visually closer to real infrared images, which preserves structural fidelity. This finding is consistent with results on the **AVIID-3** [45] datasets (Table 1 in [30]), where IR-GAN ranks next best performer after IRKDGAN. Th performance is consistent and is accessible for implementations, therefore IR-GAN is considered as the most suitable model for the synthesis of high-quality thermal images.

Method	Summary
Pix2Pix	It's a lightweight baseline that is both the smallest and the fastest among the compared methods. It performs better on **AVIID-3** [45] where it has better metrics scores (PSNR/SSIM/MS-SSIM/LPIPS) than InfraGAN. The images generated are of good quality, and it remains a good option when speed and efficiency are prioritized over fidelity.

InfraGAN	The largest and heaviest model with slowest training due to its very large discriminator. It failed behind Pix2Pix on **AVIID** [45]**.** It achieves strong perceptual results such as FID or KID, but its overall performance is inconsistent. The visual quality is less when compared with the other ranked models and is more suitable when computational resources are not constrained.

Table 6: Summary of other contenders

Table 6 illustrates a summary of other contenders for the synthesis of thermal images [30]. It is based on performance observed on the **AVIID-3** [45] datasets (Table 2 and Fig. 7 in [30]). If IR-GAN doesn't perform well when trained on RGB-thermal image pairs, then these can be used as alternatives for experimentation.

4.3.2 Data Acquisition and Preparation

The thermal and RGB data are collected using **DJI Matrice 30T** drone system equipped with DJI Zenmuse H20T sensor payload, which is a multi-camera module that includes RGB camera and an infrared (IR) thermal sensor. RGB data are collected using a wide RGB camera, while the corresponding thermal data are collected using an Infrared thermal sensor. The acquired data contains synchronous RGB and thermal video streams stored in .MP4 format. Files with suffix _W correspond to wide RGB camera video stream, whereas files with suffix _S correspond to IR thermal video stream. For each scene captured (e.g., DJI_YYYYMMDDHHMMSS_000X_*), the video streams of RGB and thermal streams represent the same scene recorded simultaneously from both RGB and thermal sensors. The preprocessing involves generating paired RGB-thermal image samples using the recorded video streams. The JPEG frames are extracted from both _W and _S .MP4 videos at matching timestamps to get RGB frames and their corresponding aligned thermal frames at the same moment in time. The extracted RGB and their corresponding thermal images are

then paired on a one-to-one basis to form the training and evaluation dataset for IR-GAN. The final preprocessing step involves cropping each paired image to 256 × 256 pixels to fit the network input requirement of IR-GAN. No further preprocessing or transformations are required on the cropped images and are directly used as training and evaluation of IR-GAN.

The authors Ma et al. have not mentioned the minimum number of samples required for training IR-GAN [31]. However, based on the number of samples used by the authors during the experimentation with IR-GAN, where they used approximately 1,000 to 1,500 samples of visible-infrared paired images of **VEDAI** [46] and **IVFG** [47] dataset for training, while around 200 to 400 samples of them were used for testing the model performance, and no separate validation split was applied on the dataset. These observations suggest that it is recommended to use 1000 or more samples for training IR-GAN, while around 200 samples can be used for testing the model performance.

4.3.3 Model Training

The preprocessed RGB–thermal image pairs are used as the input-output training data for IR-GAN. The RGB images are used as inputs to the generator, while their corresponding thermal images act as ground truth targets. The generator adopts a U-Net architecture enhanced with ConvNeXt modules, enabling more effective extraction of texture features from the RGB images. The discriminator is designed as a multi-scale PatchGAN to capture both local structural details and global contextual information from the thermal images. During training, the generator is optimized to synthesize realistic thermal images from RGB images, and the discriminator learns to distinguish between the real and the generated thermal images. The training objective combines three components: adversarial loss to encourage realism, L1 reconstruction loss weighted by λ minimizing pixel-wise deviations from the ground truth, and gradient vector loss weighted by φ to enhance edge sharpness and structural consistency. The optimization is carried out using the Adam optimizer with momentum parameters β_1 and β_2. A fixed number of epochs are employed, with a scheduled learning rate to ensure stable convergence [31]. The entire training process is conducted on a GPU to accelerate both forward and backward computations.

Hyperparameter	Value
Learning Rate	0.0002 for the first 100 epochs and decreased linearly to 0 for the remaining epochs
β_1	0.5
β_2	0.999
Epochs	200
L1 loss weight (λ)	100
Gradient Vector Loss Weight (φ)	22

Table 7: Tunable Hyperparameters for IR-GAN

Table 7 illustrates the key hyperparameters used to train IR-GAN on RGB-thermal image pairs. Each value can be tuned to improve the performance of the quantitative metrics. The overfitting can be avoided by adopting a fixed number of training epochs together with a scheduled learning rate strategy, where the rate remains constant during the initial training phase and is gradually reduced to ensure stable convergence and improved generalization.

4.3.4 Evaluation

The images generated by IR-GAN are evaluated to assess the performance of the model. The **quantitative metrics** (PSNR, SSIM, MS-SSIM, and FID in [30]) are used to assess the fidelity and quality of the synthesized thermal images. The Peak Signal-to-Noise Ratio (PSNR) is used to assess the quality of the generated thermal images by measuring the Mean Square Error (MSE) between the real thermal images and the generated thermal images by utilizing height and width of the images. A higher value of PSNR indicates smaller error between the real thermal images and the generated thermal images. The Structural Similarity Index Measure (SSIM) is used to assess the similarity between the real and generated thermal images on the basis of brightness, contrast and structural similarity. The

SSIM value is calculated by measuring means, standard deviation values and covariance of real and generated thermal images, along with the measurement of parameters for ensuring stability. The parameters are calculated using the range of image pixels. A higher SSIM value indicates that the generated thermal images have better structural consistency with the real thermal images. The Multi-Scale Structural Similarity (MS-SSIM) is used to measure the performance stability of images of different resolutions. It extends SSIM by calculating brightness, contrast and structural similarity across multiple scales. A higher MS-SSIM value indicates better multi-scale similarity and overall image quality. The Fréchet Inception Distance (FID) is used to measure similarity between the real and generated thermal images by calculating the statistical distance between feature vectors of the real and generated thermal images. A lower FID value indicates that the generated images are of high-quality and the real thermal images have a higher correlation with generated thermal images. The **qualitative evaluation** is performed by visually comparing the generated thermal images with real thermal images. This evaluation will focus on assessing if the generated thermal images by IR-GAN have clearer texture information, realistic gray information and the generated thermal images are closer to the real thermal images without artifacts.

Qualitative limitations and edge cases:

The model may struggle to map features correctly if the training samples of RGB-thermal image pairs are not perfectly aligned, which can lead to blurry thermal image outputs. It can exhibit texture information loss where temperature variations are not preserved, show edge distortion where object boundaries appear geometrically inconsistent, or produce incorrect thermal intensities that don't correspond to the expected temperature distribution in the generated thermal images.

Methodology for testing qualitative limitations:

A small set of test samples covering diverse scenarios (e.g., varying lighting and object temperatures) is selected. For each sample, the generated thermal images are compared against the corresponding real thermal images. The samples are visually inspected to check boundary sharpness, the presence of misalignments, and the logical consistency of temperature distribution relative to the objects identified in the RGB images.

4.4 RADAR Sensor Data

This section explains the methodology involved in the generation of synthetic radar images using the two-dimensional representations obtained from FMCW radar sensor measurements. The workflow contains data preprocessing and preparation steps to transform radar signals into image-like representations suitable for the generative methods. The methods are selected based on their ability to preserve structural patterns. The evaluation process ensures that the synthetic radar images show structural patterns aligned with real radar images.

4.4.1 Model Selection

The authors have explored image synthesis using radar measurements, with existing research primarily focusing on either RA (range–angle) or RD (range–Doppler) representation. The time-domain signal obtained from BGT60TR13C radar sensor can be processed into RA map containing range and angle of the target, or into RD map containing range and velocity of the detected object [4]. The aim is to identify which generative model performs best for RA image synthesis and which performs best for RD image synthesis by only utilizing radar sensor data. However, there is not much to argue about choosing the best generative model for RA image synthesis by using radar sensor data only since the only method was proposed by Fidelis et al., where WaveGAN was used to generate synthetic IF chirps conditioned on target distance, and then the generated chirps were processed into RA image by applying signal processing technique [32]. Hence, this research will focus on RD image synthesis only. The Oliveira and Bekooij proposed the Generator-Predictor system, which is a combination of a GAN and CAE [33], whereas Wachtel et al. proposed a Conditional GAN (CGAN) with deep convolutions. Wachtel et al. mentioned that their study is an improvement over the work of Oliveira and Bekooij in terms of architecture, data collection, generated data, and analysis of results [10]. A comparison between the generative frameworks will determine the more suitable method for the synthesis of RD images.

Feature	Comparison (Generator-Predictor vs CGAN)
Generative Aim	The Generator-Predictor system is used to synthesize a sequence of RD maps containing motion of an object over time based on the object's initial position, while the CGAN is used to generate RD maps of specific class by providing the generator and discriminator networks with the target class. Since CGAN gives control over the generator output explicitly, it is a viable option for the synthesis of RD images.
Architecture	The Generator-Predictor system uses GAN to produce first frame and CAE for predicting sequential frames and denoising, resulting in architecture complexity. On the other hand, CGAN utilizes a single framework for training and generation, which leads to reduction in computational costs. Since CGAN is computationally efficient over Generator-Predictor system, it can be considered as an efficient option for the synthesis of RD images.
Data Preprocessing	A preprocessing step for the RD maps before training the Generator-Predictor system involved converting the real RD images into black and white images. This alters the structure of the input data, where the synthesized RD maps representation lacks intensity variations, which is a characteristic of the actual FMCW radar output, as the original form was not restored. On the other hand, the preprocessing step before training the CGAN involved cropping the RD targets, which enables the model to focus on the relevant scattering patterns of targets. Thus, the structure is preserved, and the synthesized images appear more closely with the real RD targets.

Evaluation Results	The Generator-Predictor system is less suitable for this research because its output is evaluated based on the detection of moving objects, rather than how well it captures class-specific features. In contrast, the CGAN evaluation focuses on its data augmentation capability, where the 4,500 generated samples were used to augment the real training data, which summed up to an amount of 18061 training samples. The results show a 0.6% increase in overall classification accuracy on the real test set. Although there is only a slight improvement in the accuracy of the real test dataset, the CGAN can be used to solve the data scarcity problem through data augmentation.

Table 8: Comparison between Generator-Predictor system and CGAN

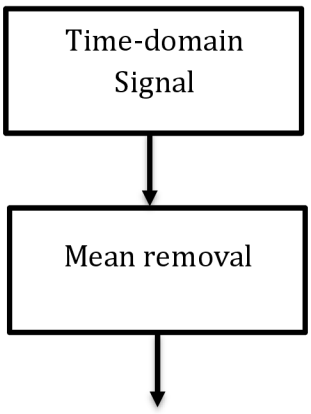
Table 8 illustrates the comparison between CGAN and Generator-Predictor system for the synthesis of RD images by comparing their features [10], [33]. It is observed that the Conditional Deep Convolutional GAN approach overcomes the architectural and methodological limitations of the Generator-Predictor system. Hence, CGAN is considered as the best possible option for the synthesis of RD images.

The CGAN is identified as the best generative model for the synthesis of RD images, but the model does not preserve radar measurable parameters such as the range or velocity of the target because the cropped RD images used for training CGAN doesn't contain the x and y (velocity and range) axes. CGAN can synthesize visually realistic RD images that can be used for augmenting training data, training classifier models, evaluating pre-trained classifier models, or qualitative validation. The assessment was performed through classifier-based validation and no other quantitative image evaluation metrics (such as FID) were applied on the generated RD images. This research focuses on synthesizing radar images using only the data obtained from radar sensors, it has been observed that the models proposed by the authors do not generate images with measurable quantities. However, by using cross-

domain translation techniques, it is possible to synthesize radar data that can be used to measure quantitative data, including RA map estimation from RGB Image, Depth, and Semantic Descriptions (RAIDS) , Radar RA Map estimation from LiDAR Data , and emulating Radar RA Point Cloud data from RGB images [48].

4.4.2 Dataset Preparation

The FMCW radar can measure the separation between the sensor and a target, the speed of a moving object, or the angle between the radar sensor and a target. The dataset is derived from a BGT60TR13C FMCW radar using an algorithmic processing chain to generate two-dimensional Range-Doppler (RD) Image. This Image is the input for the generative model. RD image is a two-dimensional representation showing ranges and velocities of the objects detected by the radar sensor. It's a two-dimensional grid with the range being represented on the y-axis and velocity being represented on the x-axis, with the range being positive values while positive and negative values for the velocity. The velocity (v) of a target is determined by the range difference (ΔR) between two consecutive chirps divided by time difference (T_I) over the period between those chirps using the formula ($v = \frac{\Delta R}{T_I}$). The velocity values being positive indicate that the object is heading away from the radar system, while negative values indicate that the object is approaching the radar sensor. The objects detected are represented using different colors of a colormap according to their strengths[4].



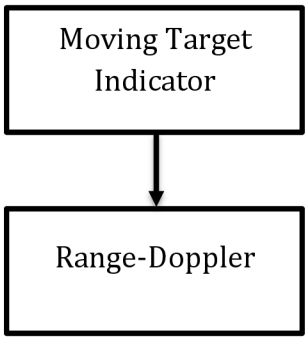


Figure 11: Algorithm processing chain based on [4]

Figure 11 illustrates the algorithm processing chain for RD map estimation. The time-domain signal is the input obtained from the BGT60TR13C radar sensor. First, the mean-value of the input signal is computed and subtracted. Next, a Moving Target Indicator (MTI) is used to eliminate static targets. Finally, from the output of the MTI, the Range-Doppler image is computed [4].

The computed RD images are used as the training dataset, which undergoes preprocessing before training the CGAN. The images are cropped to extract only the RD targets with Micro-Doppler from the complete RD map, instead of using the entire map that includes both range and velocity axes. The cropped images are then standardized to 64 × 64 pixels. If class labels are present, then these images are labeled according to the available categories. This prepared and labeled dataset serves as the ground truth for training the CGAN. However, if the class labels are not present, then the standardized images are treated as a single dataset without labels. This unlabeled dataset could be used for training the unconditional Deep Convolutional GAN (DCGAN), which learns the general distribution of all the RD images instead of specific categories.

4.4.3 Model Training

The CGAN for RD image synthesis can handle labeled datasets but will require some modifications for handling unlabeled datasets so that the generative network can either

create class-specific images or learn the overall data distribution. In this study, the model training process for both scenarios will be discussed. The labeled dataset consists of class categories to train the Conditional Deep Convolutional GAN (CGAN) where the model is trained to generate synthetic RD images conditioned on a specified class. In the CGAN, a class label is concatenated to the input of both generator and discriminator networks allowing the model to train with a condition that controls the output. Both networks are trained using the Adam optimizer, with the learning rate and number of epochs defined as tunable hyperparameters. Once the generative framework is trained and saved, it can be used for creating synthetic RD images based on the selected condition [10]. The unlabeled dataset doesn't contain class categories. The model is trained to learn the general distribution of all RD images rather than generating class-specific ones. This can be achieved by adapting the training process of the established CGAN architecture where the class label input is removed, and therefore no conditional label is concatenated to the input of either the generator or discriminator networks. The resulting model is an unconditional GAN with deep convolutional layers, commonly referred to as a Deep Convolutional GAN (DCGAN). The training process remains the same where once the model is trained and saved, it can be used to generate synthetic RD images that represent the overall characteristics of the training data.

4.4.4 Evaluation

The generated images are evaluated to ensure that the synthetic images produced by the GAN models accurately represent the real images. The evaluation strategy is applied depending on whether the generated images are labeled or unlabeled. Both qualitative and quantitative evaluation methods are applied to assess the performance of the models. The evaluation of generated RD images with labels is performed by visually comparing with the real images within the same class category to assess how well the CGAN captures the patterns. The CNN classifier trained on the real labelled dataset will be used to assess the CGAN model performance based on the generated images. A high classification accuracy indicates that the generated images are realistic and close to the real RD images data distribution. The evaluation of generated RD images without labels is performed visually by comparing the entire collection of real RD images to verify if the synthetic images resemble

the patterns in the real training samples. The FID [30] is used to measure similarity between the real and generated RD images by calculating the statistical distance between feature vectors of the real and generated RD images. A lower FID score indicates that the synthetic RD images are of high quality and have high correlation with real RD images.

Qualitative limitations and edge cases:
The model may generate very similar radar images for the same class, which indicates mode collapse. It may produce smooth backgrounds instead of the realistic noise patterns seen in the real maps. Also, the images generated may contain unrealistic Micro-Doppler patterns, which can reduce the accuracy in the classification tasks.

Methodology for testing qualitative limitations:
The synthetic images are generated for each class and visually compared with the real images to check for repetition to detect mode collapse. The background regions in the generated images are compared with real RD maps to verify the noise texture. The shape and spread of the generated images of target class are inspected to verify that meaningful Micro-Doppler patterns are present and not replaced by generic or blurred artifacts.

4.5 Laser Scanner Sensor Data

This section describes the methodology involved for the synthesis of LiDAR images using data obtained from drone-mounted laser scanners. The subsections describe the selection of a generative model suitable for range-reflectance images and the preprocessing steps required for the three-dimensional point cloud data. Finally, it outlines the model training and evaluation strategies used to ensure that the synthetic outputs are statistically aligned with the real data.

4.5.1 Model Selection

The authors Nakashima et al. evaluated various generative models for the unconditional LiDAR data generation on the **KITTI-360** [49] dataset [34]. Based on the evaluation, the most suitable models for the synthesis of laser scanner images are among the GAN-based models, Diffusion-based models, and the rectified flow-based model (R2Flow).

Rank	Method
1	R2Flow
2	R2DM
3	LiDARGen

Table 9: Ranking of the models for Laser Scanner Image Synthesis

Table 9 illustrates the ranking of the competing models for the generation of laser scanner (LiDAR) range-reflectance images. The ranking is based on the quantitative evaluation for unconditional generation on **KITTI-360** [49] (Table I, and Fig.7 in [34]) using only the best FRD metric, and qualitative evaluation. The FRD is the only evaluation metric that incorporates both range and reflectance modalities, and it is a key requirement for laser scanner (LiDAR) range-reflectance image generation, since other metrics are derived from the range image only, they are not considered in the ranking process. The iterative models that do not report FRD are also not included in the ranking and GANs exhibit very high FRD scores. The quantitative evaluation (FRD) is for assessing distributional similarity between real and generated samples, and the qualitative evaluation is for assessing scan lines and boundaries.

The R2Flow approach shows better performance for speed-quality tradeoff in terms of FRD against baseline methods (Fig. 6 in [34]), which includes GANs and Diffusion-based models. It achieves better sample quality for LiDAR data generation. On the **KITTI-360**[49] dataset (evaluated at 256 NFE, Table I in [34]), the 1-RF variant of R2Flow achieved the best FRD score, and the 2-RF variant of R2Flow maintained better performance as well. The Diffusion-based models exhibit performance degradation in FRD when NFE is reduced, whereas R2Flow mitigates this degradation through reflow and distillation, since distillation reduces FRD degradation at low NFE (Table I in [34]). The qualitative results also indicate R2Flow produces sharper scan lines and clearer object boundaries (Fig. 7 in [34]). Therefore, based on FRD at the chosen NFE, R2Flow is selected as the best model for laser scanner image synthesis.

Method	Summary
R2DM	The R2DM is a DDPM-based framework that performs pixel-space iteration and requires significant computational cost to generate high-quality LIDAR images. R2DM at 256 NFE achieves strong FRD performance, and the qualitative results indicate that, just like R2Flow, R2DM also produces sharper scan lines and clearer object boundaries. It is ranked below R2Flow because its FRD performance degrades when NFE is reduced, which reflects trade-off between quality and sampling efficiency in the iterative generation process.
LiDARGen	The LiDARGen is a SMLD-based framework that performs pixel-space iteration and is the lowest performing model among the compared iterative models based on FRD and efficiency. Because it requires a very high number of function evaluations (NFE = 1160), it results in higher computational costs than R2DM, and R2Flow. The FRD performance is worse than R2DM (with 256 NFE), and worse than R2Flow, which achieves lower FRD with fewer NFE. The qualitative results also indicate that LiDARGen generates poorer range-reflectance images than other iterative models, which makes it an inefficient choice for the laser scanner image generation.

Table 10: Summary of other contenders

Table 10 illustrates a summary of other contenders for the synthesis of laser scanner images. It is based on the FRD performance, qualitative evaluation, and computational efficiency observed on the **KITTI-360** [49] dataset (Table I, Fig. 6 and Fig.7 in [34]). If

R2Flow doesn't perform well when trained on Laser Scanner data, then other ranked models in Table 9 can be used for the synthesis of Laser scanner images. The GAN-based methods, which include Vanilla GAN, DUSty v1, and DUSty v2 have high FRD scores, the qualitative results are better for DUSty v2 but there are deviations from real data distributions. DUSty v2 performs best within the GAN group, its FRD is not competitive with the iterative models with a higher NFE. Also, Nakashima et al. mentioned that the iterative models are better than the GANs, which can suffer from unstable training, and their generated samples deviate from real data. Hence, the GAN-based models are less suitable for laser scanner (LiDAR) range-reflectance image generation based on FRD evaluation.

4.5.2 Dataset Preparation

The dataset consists of three-dimensional point clouds captured using a YellowScan Mapper, which is a UAV LiDAR-based system. The data is stored in LAS file format, where individual points in a point cloud contain three-dimensional coordinates (X, Y, and Z), and the strength of the laser return signal (reflectance) [5]. The data preparation involves projecting each point cloud into a two-channel equirectangular image representation via spherical projection, where the channels correspond to range and reflectance. The range values are converted to a log-scale representation and normalized to stabilize training. The resulting equirectangular range-reflectance images are used as training samples[34].

4.5.3 Model Training

The R2Flow model is trained using preprocessed equirectangular images. The training procedure involves two main stages as part of Rectified Flow framework. In the first stage (1-RF), the model learns the initial basic transformation paths from noise to the preprocessed equirectangular training images. The 1-RF model can produce high-quality samples, but the trajectories are curved. In the second reflow stage (2-RF), the model uses what it already learned from the first stage (1-RF model) to generate a new set of training pairs so that the model learns to create straighter paths from noise to images. The second stage uses a pseudo-Huber loss and a U-shaped timestep sampling strategy to improve training. In the final stage, to achieve extremely fast sampling (e.g., with 1, 2, or 4 NFE), the timestep distillation (k-TD) is applied, where the 2-RF model training focuses only on the

specific timesteps required for few-step sampling, which results in a model (e.g., 2-RF+k-TD) for fast and realistic generation of laser scanner (LiDAR) range-reflectance images [34].

4.5.4 Evaluation

The generated two-dimensional laser scanner (LiDAR) range-reflectance images are projected to three-dimensional point clouds using inverse of spherical projection for qualitative evaluation. The **quantitative evaluation** is performed using Fréchet Range Distance (FRD), which is used for evaluating the squared Wasserstein metric between mean and the covariance of a LiDAR perception network's activations from the synthetic samples and true samples [50]. This means it evaluates how similar the internal feature distributions are between synthetic and real LiDAR images. A lower FRD score means synthetic samples feature distribution is closer to real samples. The **qualitative evaluation** is done through visual inspection of reconstructed point clouds by examining scan lines and object boundaries to assess the R2Flow model performance for drone LiDAR data generation.

Qualitative limitations and edge cases:

The model may produce geometric distortions when the generated two-dimensional range-reflectance images are projected into three-dimensional point clouds (e.g., flat surfaces may appear wavy). There may be presence of artifacts such as missing points, incorrectly generated points, or blurred boundaries, which can lead to unrealistic geometric structures of the three-dimensional point clouds.

Methodology for testing qualitative limitations:

A sample of generated range-reflectance images are projected into three-dimensional point clouds and are visually inspected to identify geometric distortions in the point clouds such as wavy patterns. The generated samples are checked for artifacts, including blurred object boundaries or incorrect points in empty areas, to ensure the geometric structure resembles real LiDAR data.

5 Results

This chapter presents the experimental results and evaluation of the proposed generative framework. There were constraints in acquiring thermal, radar and laser scanner datasets, the evaluation is focused only on the acquired RGB camera sensor data. The results for other sensor datasets discussed in methodology are therefore not included. This chapter evaluates the performance of the SinDiffusion for RGB image synthesis using quantitative metrics and qualitative visual analysis.

5.1 RGB Camera Sensor Data

This chapter discusses the experimental results for synthetic RGB images that are obtained using the SinDiffusion model. The experiments were conducted on the real RGB images, which were captured using the **DJI Mavic 3M** drone mounted downward facing and forward facing RGB camera. Each experiment consists of training SinDiffusion on one real image and generating multiple synthetic images from that image. The evaluation contains both quantitative metrics and visual analysis to assess fidelity, diversity, and failure modes. All experiments were performed by using Python 3.11.10 version and PyTorch 2.5.0 with CUDA support on an NVIDIA Tesla V100-PCIE-32GB GPU.

5.2 Quantitative Evaluation

The quantitative evaluation of the generated RGB images is performed using SIFID and LPIPS metrics. The experimental setup generates image at a resolution of 166 × 500 pixels. The model training takes approximately 5 to 6 hours. A total of 100 synthetic images is generated for each real image, which gives 4,950 unique image pairs for LPIPS computation based on the formula $\frac{n\times(n-1)}{2}$, where n = number of synthetic images (n>1).

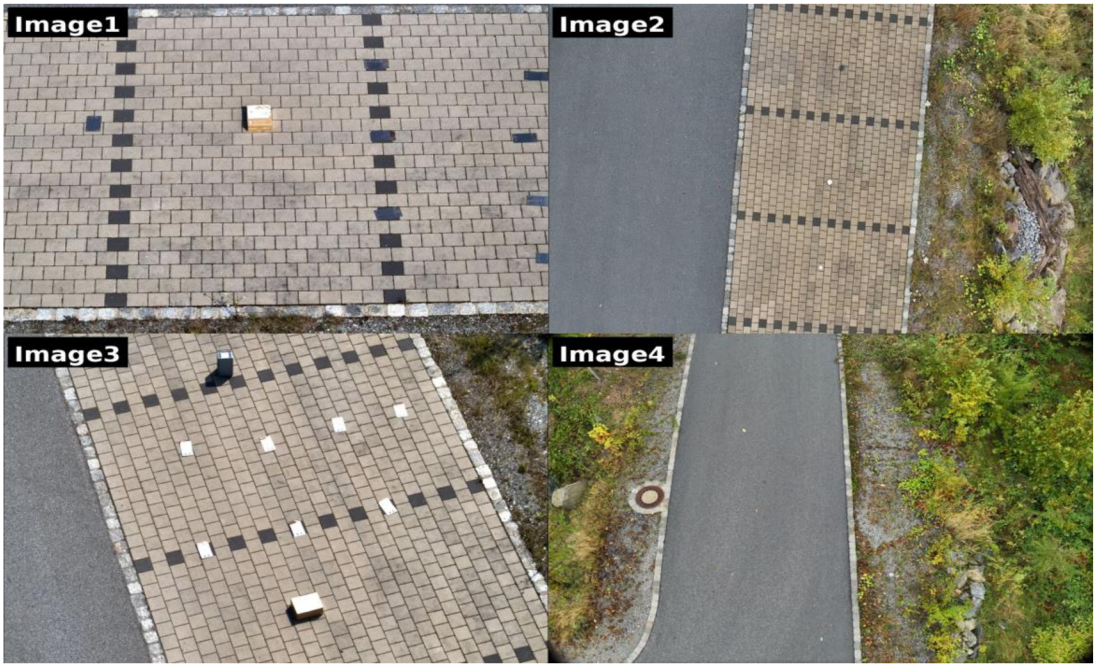


Figure 12: Downward Facing (DF) Camera Images

Figure 13: Forward Facing (FF) Camera Images

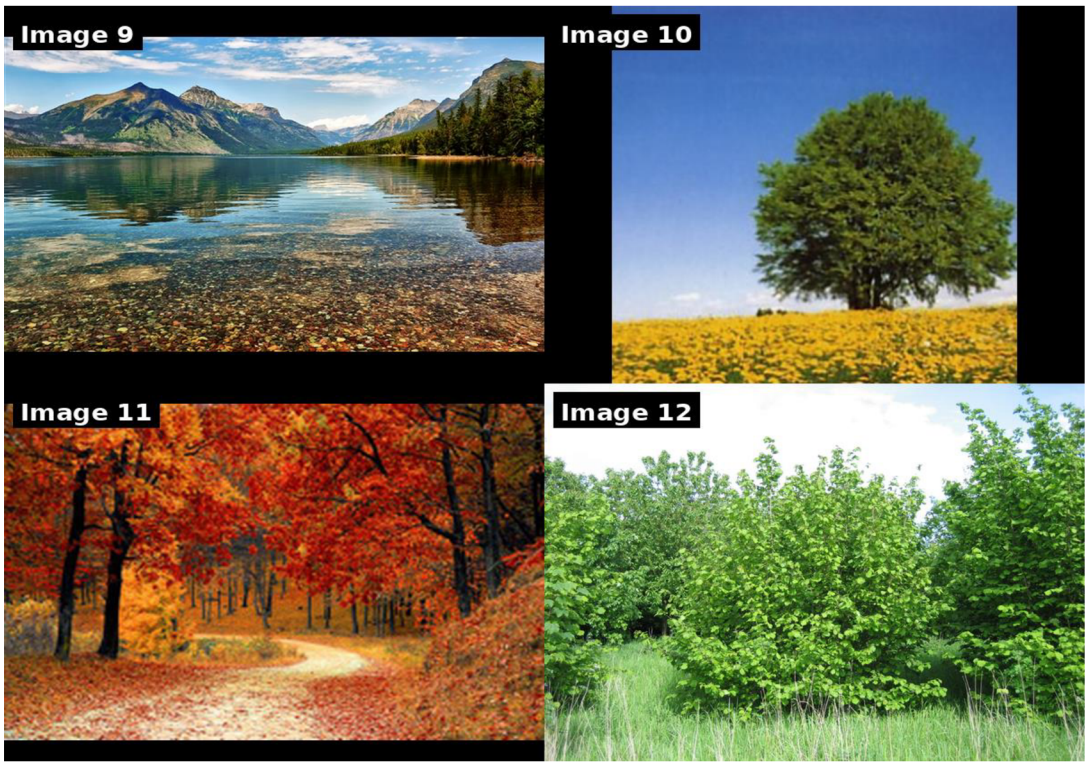


Figure 14: Real Samples from Place50 Dataset [44]

Figure 12 contains four samples of real images captured using Downward Facing (DF) camera, Figure 13 contains four samples of real images captured using Forward Facing (FF) camera. Figure 12 illustrates different ground surfaces and object placements observed from drone's perspective, Figure 13 illustrates forested landscapes and paved road environments, under different viewpoints and conditions. Figure 14 includes few samples of random images from the **Place50** [44] dataset (P50) which was used by authors Wang et al. in their research on SinDiffusion [25].

Scene View	SIFID ↓ (min, max, mean, std)	LPIPS ↑ (min, max, mean, std)
Image1 (DF)	(0.48, 1.16, 0.81, 0.13)	(0.24, 0.50, 0.36, 0.04)
Image2 (DF)	(0.21, 0.33, 0.26, 0.02)	(0.35, 0.80, 0.63, 0.06)
Image3 (DF)	(0.59, 0.88, 0.73, 0.06)	(0.15, 0.37, 0.27, 0.03)

Image4 (DF)	(0.23, 0.45, 0.32, 0.04)	(0.38, 0.78, 0.60, 0.06)
Image5 (FF)	(0.13, 0.61, 0.30, 0.10)	(0.19, 0.66, 0.44, 0.08)
Image6 (FF)	(0.13, 0.19, 0.16, 0.01)	(0.18, 0.38, 0.29, 0.03)
Image7 (FF)	(0.12, 0.24, 0.17, 0.03)	(0.28, 0.70, 0.55, 0.06)
Image8 (FF)	(0.08, 0.17, 0.14, 0.02)	(0.40, 0.64, 0.53, 0.03)
Image9 (P50)	(0.19, 0.39, 0.26, 0.04)	(0.25, 0.49, 0.39, 0.03)
Image10 (P50)	(0.67, 0.92, 0.79, 0.05)	(0.13, 0.64, 0.44, 0.08)
Image11 (P50)	(1.16, 1.55, 1.33, 0.09)	(0.37, 0.63, 0.52, 0.04)
Image12 (P50)	(0.29, 0.41, 0.36, 0.03)	(0.20, 0.36, 0.28, 0.02)

Table 11: Quantitative Results for RGB image synthesis

Table 11 presents quantitative results for RGB image synthesis using SinDiffusion, which is evaluated using SIFID (↓) to measure fidelity (lower indicates better similarity to the real image) and LPIPS (↑) to measure diversity (higher indicates greater perceptual variation among generated samples). In case of scenes containing DF images (Image 1-4), Image2 has the strongest fidelity because it achieves the lowest SIFID min, max and mean values. This indicates SinDiffusion preserves the appearance and structure of the Image2 most effectively across generated samples. Image1 has the highest SIFID max and mean, and indicates weakest fidelity among the DF scenes, which suggests perceptual deviations relative to the real image. In terms of diversity, Image2 also produces the most diverse generations among the DF scenes, which achieve the highest LPIPS max and mean. This indicates strong variation across the generated samples. Image4 has the highest LPIPS minimum, which indicates that the most similar pairs generated are distinct. The image with least diversity in DF case is Image3 because it has the lowest LPIPS min/max/mean, which means less diverse outputs. In case of Forward Facing (FF) scenes (Images 5–8), the best fidelity is achieved by Image8, which has the lowest SIFID min/max/mean values. This indicates that SinDiffusion can reproduce Image8 scene with high perceptual similarity across the generated samples and indicates the best overall fidelity result in the table. The weakest FF fidelity is observed for Image5, which has the highest SIFID max and mean. This suggests that generated images have larger deviations from the real image due to scene

complexity. For diversity, Image7 achieved the highest LPIPS and a higher mean, which indicates high variations in generated images. The least diverse FF output with the lowest LPIPS min/max/ mean is observed for Image6, which shows comparatively limited variation between generated images. In case of Place50 (P50) scenes (Images 9–12), Image9 has the best fidelity, which achieves the lowest SIFID min/max/mean values. This suggests that SinDiffusion preserves the structure of the synthesized images close to the real image for this P50 scenes. Image11 has the highest SIFID min/max/mean, this indicates loss of fidelity and mismatches in real and generated samples. For diversity, Image11 achieves the highest LPIPS minimum and mean, which means it produces highly diverse images, but coincides with the poorest fidelity. This indicates that the variation may not preserve the underlying structure completely for the real scene. Image10 has the highest LPIPS maximum, which indicates the presence of some dissimilar sample pairs, and Image12 shows the lowest diversity with the lowest LPIPS max and mean, which indicates repeated images and reduced variation.

5.3 Qualitative Evaluation

The visual comparison is performed on the quantitative findings to determine whether the synthesized images capture the patch distribution with structural consistency and if they are significantly diverse. The qualitative assessment for every scene seems not a practical choice, a list of shortlisted scenes is selected based on the SIFID and LPIPS results. The shortlisting is done based on the images that correspond to the best fidelity cases (lowest SIFID), the worst fidelity cases (highest SIFID), the most diverse cases (highest LPIPS) and the least diverse cases (lowest LPIPS) within each scene group (DF, FF). One visually better generated sample having variations from the real image is randomly selected and presented for each real image from each scenes group (DF, FF, P50) but excluding best fidelity cases (Image 2 and Image 8) for FF and DF scene group to avoid repetition. This way of selection ensures that the qualitative analysis captures both performance and failures and provides evidence whether SinDiffusion is the best model for RGB image synthesis or whether

alternative proposed generative models should be considered.

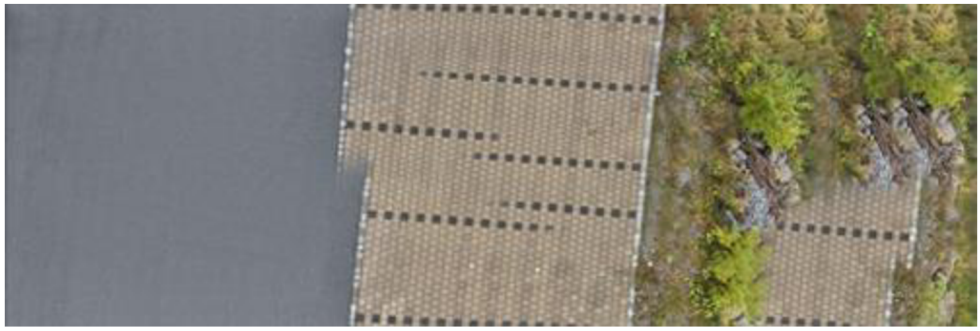


Figure 15: Best Fidelity Image from DF generation (SIFID = 0.21)

Figure 15 presents the best fidelity Image from DF generation. The generated image stays close to the real scene shown in Figure 12 (Image 2). The road, the paved surface, and the edge with vegetation and rocks are kept in the right places. The paving surface looks consistent, and the structure is well preserved. The generation indicates strong structural consistency, texture retention and supports the low SIFID result. However, there is slight merging of road and the paved surface.

Figure 16: Worst Fidelity Image from DF generation (SIFID = 1.16)

Figure 16 presents the worst fidelity image from DF generation. The generated image is realistic, and the real scene is shown in Figure 12 (Image 1). The paving color tone has changed to brown, and the block pattern looks realistic and stable and there is enough variation from Image 1 based on the black blocks. Even the edges look realistic. The model

has made a good generalization for Image 1 having repeated block patterns of the paved surface.

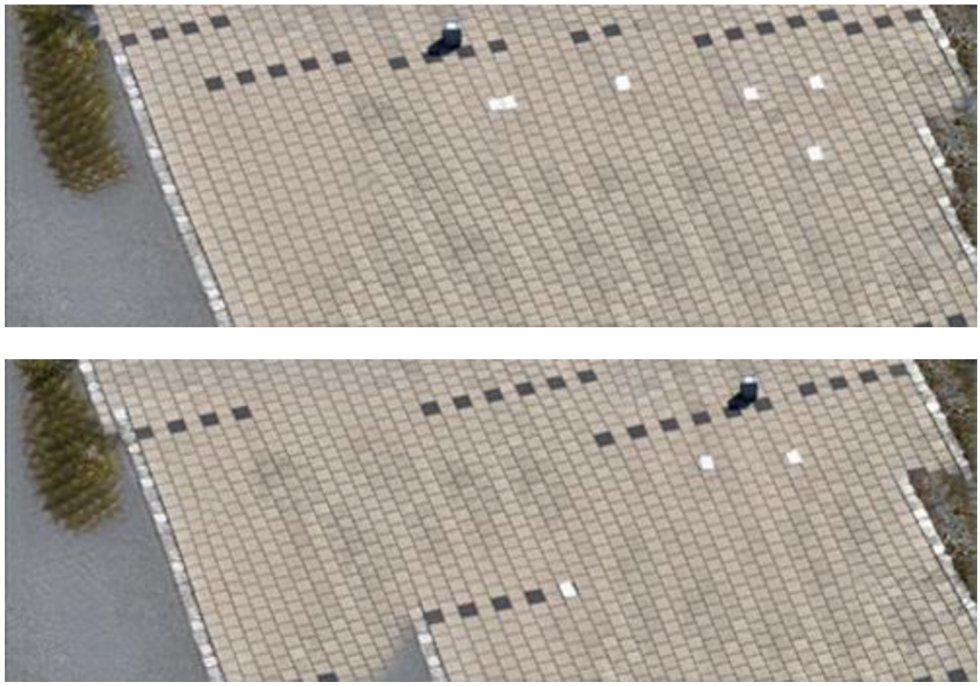


Figure 17: Least Diverse Image Pairs from DF generation (LPIPS = 0.15)

Figure 17 presents the least diverse image pairs from DF generation. The two images are generated samples for Image 3 shown in Figure 12. Thet both are visually similar, which indicate limited perceptual variation across images. The paved blocks look the same and overall structure has few variations but the position of the objects on the paved surfaces have changed and it shows variations. This is consistent with the lowest LPIPS distribution and suggests partial mode collapse due to repetitive patterns present in the image.

Figure 18: Most Diverse Image Pairs from DF generation (LPIPS = 0.80)

Figure 18 presents the most diverse image pairs from DF generation. The two images are generated samples for Image 2 shown in Figure 12. These samples show clear differences, which indicate highest perceptual variation across images, but they don't look completely realistic due to the sudden placement and merging of the road surface in the paved areas. This indicates higher diversity without completely losing the scene structure.

Figure 19: Best Fidelity Image from FF Generation (SIFID = 0.08)

Figure 19 presents the best fidelity Image from FF generation. The generated image

maintains strong structural consistency and looks visually close to the real scene shown in Figure 13 (Image 8). The road layout is clear and continuous, the vegetation on both sides has formed well, and the buildings in the background remain stable with reasonable shape and color. The edges between road, pavement, and greenery are sharp, and the overall perspective matches a realistic forward-facing drone view. This supports the low SIFID score and indicates that SinDiffusion performs best on this type of complex structured outdoor scene.

Figure 20: Worst Fidelity for FF Generation (SIFID = 0.61)

Figure 20 presents the worst fidelity image from FF generation, and the real image is shown in Figure 13 (Image 5). This output shows quality issues with blurred and fine details in the trees and terrain are not preserved completely due to merging of atmosphere and vegetation. The scene breaks structural continuity and reduces texture accuracy in some parts of the image. These artifacts are consistent with the weakest fidelity case (higher SIFID), which indicates sometimes SinDiffusion struggles to reproduce scenes reliably.

Figure 21: Least Diverse Image Pairs for FF Generation (LPIPS = 0.18)

Figure 21 presents the lease diverse image pairs from FF generation, and the real image is shown in Figure 13 (Image 6). The two forest landscape samples are almost similar and identical in structure. The tree line, horizon, and distant hills remain in the same locations with very similar lighting and color. The differences are limited to minor texture changes in the vegetation, with no clear change in scene geometric structure. This matches the low LPIPS case.

Figure 22: Most Diverse Image Pairs from FF generation (LPIPS = 0.70)

Figure 22 presents the most diverse image pairs from FF generation. The two samples show strong variation and the below sample maintain the same overall paved surface and road scene type shown in the real image in Figure 13 (Image 7). The road curvature and the paved blocks change between the samples, and the scene. This supports the high LPIPS result because the model produces noticeably different versions of the same environment, but the images seem unrealistic when there is very high diversity due to the merging of the road and the paved surfaces.

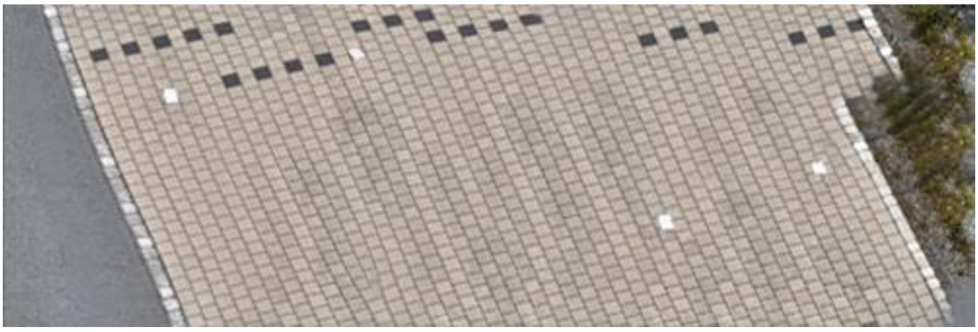


Figure 23: Generated samples for selected DF, FF, and P50 scenes.

Figure 23 presents the images containing one generated sample for each real image from Figure12 (excluding Image 4), Figure 13 (excluding Image8) and Figure 14. The visual images indicate that the samples generally preserve the overall appearance of the input and produce better visual quality across DF, FF and P50 scenes. The images reproduce local texture and patches with variations from real image. However, there are very few limitations, such as boundary misplacements and slight structural inconsistencies only on images containing roads and paved surfaces.

5.4 Model Performance Discussion

The experimental results show SinDiffusion strong performance on the drone captured scenes. The model generates better fidelity and highly diverse images with several real images from DF and FF scene groups by achieving both high fidelity and meaningful diversity (Fig. 23). The quantitative metrics (Table 11) scores indicate that the model is very effective when the scene contains consistent textures. From DF scenes group, Image2

achieves the strongest fidelity case, and for FF scene group Image8 achieves the best performing fidelity case and the best overall fidelity result among all the scenes. The diversity is not high across all drone scenes, where some views produce near duplicates (Fig. 21), which indicates mode collapse or limited variation when the underlying pattern is repetitive. The qualitative results also align with the metrics scores, where the best fidelity for DF and FF examples preserve the road placement, paved regions, vegetation boundaries (Fig. 15, 19) and maintain patch distribution with low SIFID scores (Fig. 15). However, some of the generated images containing ground surfaces show limitations of SinDiffusion where the structured environments with roads and pavements sometimes show boundary misplacements (3rd Image in Fig. 23), merging artifacts (6th Image in Fig. 23) are visible. The quantitative scores also support this where the weakest fidelity scenes for FF scene (5th Image in Table 11 and Fig. 20) are visually poor.

The DF and FF scenes compared against the Place50 (P50) images show that Images containing ground surfaces (DF/FF) have issues with misalignment and merging of road and paved surfaces because these scenes have strict boundaries that must stay in the exact position (2nd, 3rd, and 6th Image in Fig. 23). The model achieves better fidelity quantitatively on DF/FF (Image2 and Image8 in Table 11 with lower SIFID corresponds to Fig. 15 and Fig. 19) but qualitatively even a small extension during generation makes the road merge into the pavement or shifts in the tile blocks, which causes merging or boundary errors. This is because the model is trained to learn internal patch distributions due to which the model can reproduce the local textures well but may not keep the exact boundary alignment. The frontal scenes in P50 and FF scenes (Fig.14) are visually more natural, and small structural changes are not easy to address, while the surface scenes captured from the drone in DF and FF images have boundary errors based on the qualitative evaluation.

5.5 Limitations

The qualitative evaluation results show limitations of SinDiffusion for RGB image synthesis. The model has difficulties in preserving structural scenes of the ground surfaces in outdoor environments, where the road surface extends into paved surface or tile block shifts at the

boundaries. These errors are visible even when the overall texture quality is high which indicates that strong local patch distributions can have minor structural inconsistency at the scene level. There are scenes having low diversity scores due to which there are minimal variations, and the images look duplicated. The boundary errors are mainly noticeable in scenes containing surface scenes captured from the drone in DF and FF images.

6 Conclusion and Future Scope

6.1 Conclusion

This thesis has investigated various generative models for the synthesis of images using the data collected from drone-mounted sensor modalities. The focus is primarily on an RGB camera, an infrared camera, a RADAR sensor, and a laser scanner. The review of various generative models has been conducted to find a suitable model for different sensor modalities based on data characteristics, training stability, fidelity, diversity, and computational efficiency. The SinDiffusion framework has been selected for the synthesis of RGB images due to its high performance when there is limited data available. The results from the experiments have shown that SinDiffusion has learned internal patch distributions from a single real RGB image successfully and has generated visually realistic and diverse synthetic images without requiring large number of training samples to train them. The quantitative evaluation using SIFID and LPIPS metrics has indicated that the model achieves better balance between fidelity and diversity for scenes captured and the patch distributions and scene layouts are more consistent in frontal scenes from the drones. The analysis of qualitative results has shown that SinDiffusion preserves textures with repetitive patterns and overall scene layouts in the generated images with frontal scenes from the drones, but the structural distortions are observed in scenes containing ground surface which contain abrupt transitions between multiple surface types. These observations have indicated geometric inconsistencies. The thesis also proposed methodologies for thermal image synthesis using IR-GAN, RADAR RD image synthesis using CGAN, and LiDAR range-reflectance image generation using R2Flow. Overall, this research has shown that generative methods can synthesize images for drone-mounted sensor captured images. The results have indicated that RGB images with high quality can be generated when there is scarcity in collecting large amounts of images. This reduces the dependency on costly and time-consuming data collection using drone mounted sensors. The study has also shown the importance of combining quantitative metrics with qualitative analysis to accurately assess the fidelity and usability of synthetic images.

6.2 Future Scope

The qualitative evaluation shows limitations in maintaining geometric consistency and structural accuracy in scenes containing ground surfaces. The generated models need to be designed which perform well on ground surface images containing multiple types of surfaces that are captured from drones. The proposed generative models for thermal, RADAR and laser scanner sensor data are not experimentally validated so the implementation, training and evaluation of IR-GAN for thermal image synthesis, CGAN for RADAR RD image synthesis, and R2Flow for LiDAR range-reflectance image generation using data acquired from drone-mounted sensor modality needs to be performed. The evaluation of these models would enable us to analyze the performance and the relevance of the proposed methodologies. The cross-modal image generation techniques need to be investigated, such as generating RADAR RD images or LiDAR image representations from RGB images. This approach can reduce the dependency on sensor data that is costly or difficult to acquire. However, a limited amount of real data samples would still be required for training and evaluating the generative models to ensure the outputs are meaningful.

7 Bibliography

[1] D. Feng *et al.*, "Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1341–1360, Mar. 2021, doi: 10.1109/TITS.2020.2972974.

[2] Z. Song *et al.*, "Synthetic Datasets for Autonomous Driving: A Survey," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 1847–1864, Jan. 2024, doi: 10.1109/TIV.2023.3331024.

[3] Z. Zhang and L. Zhu, "A Review on Unmanned Aerial Vehicle Remote Sensing: Platforms, Sensors, Data Processing Methods, and Applications," *Drones*, vol. 7, no. 6, Art. no. 6, Jun. 2023, doi: 10.3390/drones7060398.

[4] "AN155322: Digital Beamforming using DEMO BGT60TR13C Radar Sensor." Accessed: Oct. 10, 2025. [Online]. Available: https://www.infineon.com/assets/row/public/documents/24/42/infineon-an155322-digital-beamforming-using-the-demo-bgt60tr13c-radar-sensor-applicationnotes-en.pdf?fileId=8ac78c8c8779172a0187e2848e2d0f73

[5] "LiDAR Point Clouds: Basics for 3D Mapping by Yellowscan," YellowScan. Accessed: Oct. 30, 2025. [Online]. Available: https://www.yellowscan.com/knowledge/LiDAR-point-cloud-basics/

[6] S. Feuerriegel, J. Hartmann, C. Janiesch, and P. Zschech, "Generative AI," *Bus. Inf. Syst. Eng.*, vol. 66, no. 1, pp. 111–126, Feb. 2024, doi: 10.1007/s12599-023-00834-7.

[7] K. S. Kaswan, J. S. Dhatterwal, K. Malik, and A. Baliyan, "Generative AI: A Review on Models and Applications," in *2023 International Conference on Communication, Security and Artificial Intelligence (ICCSAI)*, Nov. 2023, pp. 699–704. doi: 10.1109/ICCSAI59793.2023.10421601.

[8] S. Vivekananthan, "Comparative Analysis of Generative Models: Enhancing Image Synthesis with VAEs, GANs, and Stable Diffusion," Aug. 16, 2024, *arXiv*: arXiv:2408.08751. doi: 10.48550/arXiv.2408.08751.

[9] I. J. Goodfellow *et al.*, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2014. Accessed: May 26, 2025. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/hash/f033ed80deb0234979a61f95710dbe25-Abstract.html

[10] D. Wachtel, S. Queiroz, T. Schön, W. Huber, and L. Faria, "A novel Conditional Generative Adversarial Networks for Automotive Radar Range-Doppler Targets Synthetic Generation," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, Sep. 2023, pp. 3964–3969. doi: 10.1109/ITSC57777.2023.10422067.

[11] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," Nov. 06, 2014, *arXiv*: arXiv:1411.1784. doi: 10.48550/arXiv.1411.1784.

[12] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," Jan. 07, 2016, *arXiv*: arXiv:1511.06434. doi: 10.48550/arXiv.1511.06434.

[13] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice: IEEE, Oct. 2017, pp. 2242–2251. doi: 10.1109/ICCV.2017.244.

[14] W. Su, H. Ye, S.-Y. Chen, L. Gao, and H. Fu, "DrawingInStyles: Portrait Image Generation and Editing With Spatially Conditioned StyleGAN," *IEEE Trans. Vis. Comput. Graph.*, vol. 29, no. 10, pp. 4074–4088, Oct. 2023, doi: 10.1109/TVCG.2022.3178734.

[15] H. Wang *et al.*, "Simultaneous restoration and super-resolution GAN for underwater image enhancement," *Front. Mar. Sci.*, vol. 10, Jun. 2023, doi: 10.3389/fmars.2023.1162295.

[16] Y. Peng, "A Comparative Analysis Between GAN and Diffusion Models in Image Generation," *Trans. Comput. Sci. Intell. Syst. Res.*, vol. 5, pp. 189–195, Aug. 2024, doi: 10.62051/0f1va465.

[17] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2020, pp. 6840–6851. Accessed: May 26, 2025. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html

[18] A. Q. Nichol and P. Dhariwal, "Improved Denoising Diffusion Probabilistic Models," in *Proceedings of the 38th International Conference on Machine Learning*, PMLR, Jul. 2021, pp. 8162–8171. Accessed: Jan. 16, 2026. [Online]. Available: https://proceedings.mlr.press/v139/nichol21a.html

[19] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA: IEEE, Jun. 2022, pp. 10674–10685. doi: 10.1109/CVPR52688.2022.01042.

[20] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," Dec. 10, 2022, *arXiv*: arXiv:1312.6114. doi: 10.48550/arXiv.1312.6114.

[21] S. Kuang, "Matrix-Variate Beta Variational Autoencoder," in *2024 6th International Conference on Communications, Information System and Computer Engineering (CISCE)*, May 2024, pp. 1309–1312. doi: 10.1109/CISCE62493.2024.10653087.

[22] T. Zhao, R. Zhao, and M. Eskenazi, "Learning Discourse-level Diversity for Neural Dialog Models using Conditional Variational Autoencoders," Oct. 21, 2017, *arXiv*: arXiv:1703.10960. doi: 10.48550/arXiv.1703.10960.

[23] M. G. Sadek, A. Y. Hassan, T. O. Diab, and A. Abdelhafeez, "Creating Images with Stable Diffusion and Generative Adversarial Networks," *Int. J. Telecommun.*, vol. 04, no. 02, pp. 1–14, Dec. 2024, doi: 10.21608/ijt.2024.329477.1064.

[24] M. Yang and Z. Wang, "Image Synthesis Under Limited Data: A Survey and Taxonomy," *Int. J. Comput. Vis.*, vol. 133, no. 6, pp. 3689–3726, Jun. 2025, doi: 10.1007/s11263-025-02357-y.

[25] W. Wang *et al.*, "SinDiffusion: Learning a Diffusion Model From a Single Natural Image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 5, pp. 3412–3423, May 2025, doi: 10.1109/TPAMI.2025.3532956.

[26] W. Wang *et al.*, "SinDiffusion: Learning a Diffusion Model from a Single Natural Image," Nov. 22, 2022, *arXiv*: arXiv:2211.12445. doi: 10.48550/arXiv.2211.12445.

[27] P. L. Suárez and A. D. Sappa, "Thermal Image Synthesis: Bridging the Gap Between Visible and Infrared Spectrum," in *Advances in Visual Computing*, G. Bebis, V. Patel, J. Gu, J. Panetta, Y. Gingold, K. Johnsen, M. S. Arefin, S. Dutta, and A. Biswas, Eds., Cham: Springer Nature Switzerland, 2025, pp. 384–396. doi: 10.1007/978-3-031-77392-1_29.

[28] M. A. Khan, "Visible to Thermal image Translation for improving visual task in low light conditions," Nov. 09, 2023, *arXiv*: arXiv:2310.20190. doi: 10.48550/arXiv.2310.20190.

[29] J. Han, M. Shoeiby, L. Petersson, and M. A. Armin, "Dual Contrastive Learning for Unsupervised Image-to-Image Translation," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2021, pp. 746–755. doi: 10.1109/CVPRW53098.2021.00084.

[30] D. Ma, J. Su, Y. Xian, and S. Li, "Discriminator guided visible-to-infrared image translation," *Complex Intell. Syst.*, vol. 11, no. 4, p. 208, Mar. 2025, doi: 10.1007/s40747-025-01827-7.

[31] D. Ma, Y. Xian, B. Li, S. Li, and D. Zhang, "Visible-to-infrared image translation based on an improved CGAN," *Vis. Comput.*, vol. 40, no. 2, pp. 1289–1298, Feb. 2024, doi: 10.1007/s00371-023-02847-5.

[32] E. C. Fidelis *et al.*, "Generation of Realistic Synthetic Raw Radar Data for Automated Driving Applications using Generative Adversarial Networks," Aug. 08, 2023, *arXiv*: arXiv:2308.02632. doi: 10.48550/arXiv.2308.02632.

[33] M. L. L. de Oliveira and M. J. G. Bekooij, "Generating Synthetic Short-Range FMCW Range-Doppler Maps Using Generative Adversarial Networks and Deep Convolutional Autoencoders," in *2020 IEEE Radar Conference (RadarConf20)*, Sep. 2020, pp. 1–6. doi: 10.1109/RadarConf2043947.2020.9266348.

[34] K. Nakashima, X. Liu, T. Miyawaki, Y. Iwashita, and R. Kurazume, "Fast LiDAR Data Generation with Rectified Flows," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, May 2025, pp. 10057–10063. doi: 10.1109/ICRA55743.2025.11127894.

[35] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015, doi: 10.1007/s11263-015-0816-y.

[36] "CIFAR-10 and CIFAR-100 datasets." Accessed: Jan. 27, 2026. [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html

[37] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep Learning Face Attributes in the Wild".

[38] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks".

[39] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "StarGAN v2: Diverse Image Synthesis for Multiple Domains," Apr. 26, 2020, *arXiv*: arXiv:1912.01865. doi: 10.48550/arXiv.1912.01865.

[40] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop," Jun. 04, 2016, *arXiv*: arXiv:1506.03365. doi: 10.48550/arXiv.1506.03365.

[41] M.-E. Nilsback and A. Zisserman, "Automated Flower Classification over a Large Number of Classes," in *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, Dec. 2008, pp. 722–729. doi: 10.1109/ICVGIP.2008.47.

[42] M.-Y. Liu *et al.*, "Few-Shot Unsupervised Image-to-Image Translation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South): IEEE, Oct. 2019, pp. 10550–10559. doi: 10.1109/ICCV.2019.01065.

[43] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "VGGFace2: A Dataset for Recognising Faces across Pose and Age," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, May 2018, pp. 67–74. doi: 10.1109/FG.2018.00020.

[44] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 Million Image Database for Scene Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1452–1464, Jun. 2018, doi: 10.1109/TPAMI.2017.2723009.

[45] Z. Han, Z. Zhang, S. Zhang, G. Zhang, and S. Mei, "Aerial Visible-to-Infrared Image Translation: Dataset, Evaluation, and Baseline," *J. Remote Sens.*, vol. 3, p. 0096, Jan. 2023, doi: 10.34133/remotesensing.0096.

[46] S. Razakarivony and F. Jurie, "Vehicle detection in aerial imagery : A small target detection benchmark," *J. Vis. Commun. Image Represent.*, vol. 34, pp. 187–203, Jan. 2016, doi: 10.1016/j.jvcir.2015.11.002.

[47] B. Li, Y. Xian, J. Su, D. Q. Zhang, and W. L. Guo, "I-GANs for Infrared Image Generation," *Complexity*, vol. 2021, no. 1, p. 6635242, 2021, doi: 10.1155/2021/6635242.

[48] P. A. Rangaraj, T. Alkanat, and A. Pandharipande, "RAIDS: Radar Range-Azimuth Map Estimation From Image, Depth, and Semantic Descriptions," *IEEE Sens. J.*, vol. 25, no. 7, pp. 12381–12388, Apr. 2025, doi: 10.1109/JSEN.2025.3540660.

[49] Y. Liao, J. Xie, and A. Geiger, "KITTI-360: A Novel Dataset and Benchmarks for Urban Scene Understanding in 2D and 3D," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3292–3310, Mar. 2023, doi: 10.1109/TPAMI.2022.3179507.

[50] V. Zyrianov, X. Zhu, and S. Wang, "Learning to Generate Realistic LiDAR Point Clouds," in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., Cham: Springer Nature Switzerland, 2022, pp. 17–35. doi: 10.1007/978-3-031-20050-2_2.