UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Data Generation and Machine Learning-Based Image Restoration in 3D Synthetic Environments

Diploma Thesis

Garyfalia Georgitziki

Supervisor: Christos Antonopoulos

July 2025

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Data Generation and Machine Learning-Based Image Restoration in 3D Synthetic Environments

Diploma Thesis

Garyfalia Georgitziki

Supervisor: Christos Antonopoulos

July 2025

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Δημιουργία Δεδομένων και Αποκατάσταση Εικόνας βασισμένη στη Χρήση Μηχανικής Μάθησης σε Συνθετικά Τρισδιάστατα Περιβάλλοντα

Διπλωματική Εργασία

Γαρυφαλιά Γεωργιτζίκη

Επιβλέπων: Χρήστος Αντωνόπουλος

Ιούλιος 2025

Approved by the Examination Committee:

Supervisor **Christos Antonopoulos**
Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **Spyros Lalis**
Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **Nikolaos Bellas**
Professor, Department of Electrical and Computer Engineering, University of Thessaly

Acknowledgements

First of all, I would like to express my deepest gratitude to my supervisor, Professor Christos Antonopoulos, for his invaluable guidance, support, and encouragement throughout the course of this thesis. His expertise, constructive feedback, and constant motivation have been instrumental in shaping the direction of this work. I am deeply thankful for the time that he invested in guiding me through the research process and for the resources he provided, which were essential to the success of this project.

I am also grateful to Professor Nikolaos Bellas and Spyros Lalis for being members of the examination committee for my thesis. Their insightful feedback and suggestions were greatly appreciated.

Most importantly, I would like to express my gratitude to my family and friends for their consistent support and understanding throughout all my academic years. Their encouragement and trust in my abilities have been greatly appreciated.

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

«Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I also declare that the results of the work have not been used to obtain another degree. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism».

The declarant

Garyfalia Georgitziki

Diploma Thesis

Data Generation and Machine Learning-Based Image Restoration in 3D Synthetic Environments

Garyfalia Georgitziki

Abstract

This thesis focuses on the application of machine learning techniques, specifically Generative Adversarial Networks (GANs), to the problem of lens flare removal in RGB images. Lens flare, an optical artifact caused by strong light sources, often degrades image quality and can obscure critical visual information. The aim of this research was to develop an automated solution to restore flare-affected images, particularly in the context of precision agricultural.

To address this challenge, synthetic datasets were generated using Unreal Engine, simulating a range of flare artifacts under varying environmental conditions. A pre-existing GAN architecture, SpA-GAN, originally designed for cloud removal, was adapted and trained to remove lens flare. The model was evaluated using quantitative metrics such as the Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and the Green-Red Vegetation Index (GRVI), ensuring that vegetation features were preserved during the restoration process.

The results demonstrate that the trained model effectively reduces flare artifacts and restores image quality while maintaining the structural integrity of the original image. This research provides a foundation for applying machine learning techniques to lens flare removal in precision agriculture and environmental monitoring, as well as in other fields.

Keywords:

Lens Flare Removal, Generative Adversarial Networks (GANs), Image Restoration, Synthetic Dataset Generation, Agricultural Image Processing, Deep Learning for Image Enhancement

Διπλωματική Εργασία

Δημιουργία Δεδομένων και Αποκατάσταση Εικόνας βασισμένη στη Χρήση Μηχανικής Μάθησης σε Συνθετικά Τρισδιάστατα Περιβάλλοντα

Γαρυφαλιά Γεωργιτζίκη

Περίληψη

Αυτή η διπλωματική εργασία επικεντρώνεται στην εφαρμογή τεχνικών μηχανικής μάθησης, συγκεκριμένα των Generative Adversarial Networks (GANs), στο πρόβλημα της αφαίρεσης αντανάκλασης φακού (lens flare) από RGB εικόνες. Η αντανάκλαση φακού, ένα οπτικό φαινόμενο που προκαλείται από έντονες πηγές φωτός, συχνά υποβαθμίζει την ποιότητα της εικόνας και μπορεί να καλύψει κρίσιμες οπτικές πληροφορίες. Σκοπός αυτής της έρευνας ήταν η ανάπτυξη μιας αυτοματοποιημένης λύσης για την αποκατάσταση των εικόνων που επηρεάζονται από αντανάκλαση φακού, ιδιαίτερα στο πλαίσιο της γεωργίας ακριβείας.

Για την αντιμετώπιση αυτής της πρόκλησης, δημιουργήθηκαν συνθετικά σύνολα δεδομένων χρησιμοποιώντας το Unreal Engine, προσομοιώνοντας διάφορα είδη αντανάκλασης σε διάφορες περιβαλλοντικές συνθήκες. Μια προϋπάρχουσα αρχιτεκτονική GAN, το SpA-GAN, που είχε αρχικά σχεδιαστεί για την αφαίρεση σύννεφων, προσαρμόστηκε και εκπαιδεύτηκε για να αφαιρεί την αντανάκλαση φακού. Το μοντέλο αξιολογήθηκε χρησιμοποιώντας μετρικές όπως ο Δείκτης Σχέσης Σήματος προς Θόρυβο (PSNR), ο Δείκτης Δομικής Ομοιότητας (SSIM) και ο Δείκτης Βλάστησης Πράσινου-Κόκκινου (GRVI), διασφαλίζοντας ότι τα χαρακτηριστικά της βλάστησης διατηρήθηκαν κατά τη διάρκεια της διαδικασίας αποκατάστασης.

Τα αποτελέσματα δείχνουν ότι το εκπαιδευμένο μοντέλο μειώνει αποτελεσματικά τα φαινόμενα αντανάκλασης και αποκαθιστά την ποιότητα της εικόνας, διατηρώντας παράλληλα τη δομική ακεραιότητα της αρχικής εικόνας. Η έρευνα αυτή παρέχει μια βάση για την εφαρμογή τεχνικών μηχανικής μάθησης στην αφαίρεση αντανάκλασης φακού στη γεωργία ακριβείας αλλά και σε άλλους τομείς.

Λέξεις-κλειδιά:

Αφαίρεση Αντανάκλασης Φακού, Generative Adversarial Networks (GANs), Αποκατάσταση

Εικόνας, Δημιουργία Συνθετικών Συνόλων Δεδομένων, Επεξεργασία Εικόνας στη Γεωργία, Βαθιά Μάθηση για Βελτίωση Εικόνας

Table of contents

List of figures

List of tables

Abbreviations

GANs	Generative Adversarial Networks
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index
MSE	Mean Squared Error
VGG	Visual Geometry Group
GRVI	Green-Red Vegetation Index
GPU	Graphics Processing Unit
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
SSD	Solid State Drive
RAM	Random Access Memory

Chapter 1

Introduction

Image capture in outdoor environments is often affected by lens flare, an optical artifact caused by strong light sources like the sun entering the camera at an angle. This results in bright spots, streaks, or haze that degrade image quality.

Lens flare is more than a visual nuisance. In practical applications such as precision agriculture, environmental monitoring, or autonomous navigation, it can obscure important elements of the scene, impairing analysis and decision-making. Traditional image processing struggles with this problem because of the flare's unpredictable shape, intensity, and location.

A key challenge is that flare can completely overwrite pixel values, making it impossible to recover hidden content without advanced reconstruction techniques. Recent advances in machine learning, and in particular deep learning, have shown great promise in addressing such complex image restoration tasks by learning to reconstruct missing or corrupted content from data.

Generative adversarial networks (GANs), a powerful class of deep learning models, have demonstrated impressive results in image-to-image translation problems, including rain removal, haze removal, and cloud removal. This thesis investigates the application of GAN-based methods for lens flare removal, leveraging synthetic datasets generated in a virtual environment to train and evaluate restoration models.

Figure 1.1: Lens Flare [1].

1.1 Scope of the Thesis

This thesis specifically addresses the problem of lens flare removal from single RGB images. The goal is to develop a robust and automated solution that can eliminate flare artifacts and restore the occluded content in a visually coherent manner.

The main problems tackled in this work include the following:

- The detection and localization of flare-affected regions, which are often irregular and lack a consistent pattern.

- Reconstruction of image content in areas where pixel data have been overwritten or obscured by flare, making the restoration process highly ill-posed.

To address these problems, the proposed approach leverages GANs, a class of deep learning models well-suited for image restoration tasks. GANs can learn to generate realistic image content by modeling complex distributions, which allows them to fill in missing or corrupted regions in a plausible way. The model is trained on a dataset containing paired synthetically generated images with and without flare, enabling it to learn the mapping from flare-degraded to clean images.

1.2 Contribution

The main contributions of this diploma thesis are outlined below:

1. A detailed study of the impact of lens flare on image quality and the unique challenges it poses for image restoration.

2. The adaptation of an existing GAN-based architecture originally designed for cloud removal [5] to the task of lens flare removal, including modifications to the training process to improve performance on flare-specific artifacts.

3. The preparation of a training dataset containing synthetic paired images with and without lens flare, designed to simulate realistic environmental conditions.

4. The evaluation of the performance of the model using quantitative metrics such as PSNR, SSIM and GRVI, together with qualitative visual analysis.

5. The development of a real-time image restoration system, enabling live inference and visualization of flare removal results. This implementation was used to test the GAN's applicability in real-time scenarios.

A video demonstration of the real-time flare removal system can be viewed at the following link: **Real-Time Flare Removal Demo using GAN**.

1.3 Thesis Structure

This thesis is organized into five main chapters. Chapter 1 introduces the scope and contributions of the work and outlines the overall structure of the thesis. Chapter 2 provides the necessary background, covering the phenomenon of image degradation caused by lens flare, an overview of GANs and attention mechanisms, related research in image restoration, and the software tools and libraries used. Chapter 3 details the implementation aspects, including dataset generation using Unreal Engine, the design of the GAN architecture, and the training procedures. This chapter also discusses preliminary training performed on a cloud removal dataset, followed by the main training on the synthetic flare dataset, as well as challenges and adaptations encountered during development. Chapter 4 presents the evaluation of the model through both quantitative metrics such as PSNR and SSIM, and qualitative visual assessments, concluding with an analysis of the model's performance. Finally, Chapter 5

summarizes the conclusions drawn from this research effort and outlines potential directions for future work.

Chapter 2

Background

2.1 Image Degradation Due to Lens Flare

Lens flare is a visual artifact that happens when strong light sources, like the sun, directly hit the lens of a camera at certain angles. This optical aberration causes an interference pattern that disrupts the cluster of wanted image points. Usually, these distortions come in the form of bright spots, streaks, or some kind of haze that causes the color and sharpness of an image to be decreased. The size of the flare is most of the time dependent on the location of the light and the lens' quality, with some lenses being more vulnerable to the flare because of their structure and optical properties [6].

Lens flare artifacts manifest in two primary forms: specular and diffuse reflections. Specular flare appears as bright hotspots, streaks, or geometric patterns that create visually prominent artifacts such as luminous blobs or radial streaking patterns across the image. Diffuse flare, conversely, manifests as a gradual reduction in image contrast, creating hazy or washed-out regions that subtly degrade overall image quality. According to [7], diffuse flare presents greater challenges for removal algorithms because it affects large image regions with subtle intensity variations that can be difficult to distinguish from legitimate scene content. If flare is present, it can cover or hide the core of the image creating the effect of the blurred image and the potential loss of the details of the scene.

The impact of lens flare is particularly problematic in applications demanding precise visual analysis. One example is computer vision which is applied to autonomous vehicle navigation, where flare can cause the system to be unable to see road signs, pedestrians, or obstacles that could lead to safety issues. Similarly, in aerial imaging for agricultural mon-

itoring or environmental assessment, flare artifacts can mask vegetation indices or terrain features essential for accurate analysis [8]. Research by [9] demonstrates that lens flare introduces significant variability in brightness and spatial distribution, making it challenging for conventional image processing algorithms to achieve consistent removal across different imaging conditions.

A primary issue occurs when a strong flare totally floods the pixel values thus, the information of the scene is lost. In such cases, the original image data becomes irrecoverable through simple processing techniques, as the dynamic range limitations of imaging sensors result in permanent information loss. Non-traditional approaches like histogram equalization, spatial filtering, and edge enhancement are not sufficient to solve the problem of the severe flare artifacts [7]. This limitation has motivated the development of advanced computational methods, including machine learning-based restoration techniques that can learn complex mappings between flared and clean images to achieve more effective artifact removal [10].

2.2 Machine Learning

Machine learning is a division of artificial intelligence (AI) that concentrates on innovating algorithms that have the ability to learn from data and make predictions or decisions. In a traditional computing model, rules are explicitly programmed whereas, machine learning systems can improve their performance as they are exposed to more data. These systems rely on statistics for identifying the patterns in the data, thus empowering them to predict, classify, or even suggest actions, without being explicitly programmed for each scenario.

At its core, machine learning involves feeding a system with data and allowing it to learn patterns or relationships that it can later apply to new, unseen data. There are several types of machine learning techniques, including supervised learning, unsupervised learning, and reinforcement learning. Each type is used based on the problem at hand and the nature of the data available.

In supervised learning, the model is trained on a labeled dataset, meaning the data includes both the input features and the correct output labels. The goal is to learn a mapping from inputs to outputs. On the contrary, unsupervised learning is the process of unlabeled data, whereby the model endeavors to discover the structure or the patterns that lie hidden in the data, such as clustering. Reinforcement learning, on the other hand, involves learning through

interaction with an environment, where the system receives feedback through rewards or penalties, driving it to improve its decision-making over time.

Machine learning has become an essential tool in many industries. Those include healthcare, finance, retail, and technology. It has allowed driving innovations such as self-driving vehicles, recommendation engines, and personalized medical treatments. With the progression of technology, machine learning is still in the process of development and deep learning, which is a part of machine learning, is the main driver for the successful experiments in the areas of image and speech recognition [11].

2.3 Deep Learning

Deep learning is a subfield of machine learning, which mainly focuses on utilizing multi-layered neural networks to represent complex patterns in data. The networks mentioned as deep neural networks are aimed to mimic the human brain's elaborate decision-making processes. Traditional machine learning models typically use one or two layers, while deep learning models generally have three or more layers—sometimes even hundreds or thousands—to carry out the steps of data transformation.

The most remarkable aspect of deep learning is its capacity to process unstructured data, such as images, audio, and text, without the assistance of a human in feature extraction. Deep learning models accomplish this by hierarchically representing the data, thus they can automatically identify relevant features at various levels of abstraction. To illustrate, in computer vision tasks, the initial layers can find edges and textures, whereas the highest layers will be able to identify objects or persons.

This capability has led to significant advancements in various fields, including computer vision, natural language processing, and speech recognition. The spectrum of deep learning-powered applications goes from voice assistants, self-driving cars to medical image analysis and fraud detection [12].

In the context of this thesis, deep learning, and specifically Generative Adversarial Networks (GANs), are applied to address the challenge of lens flare removal in images. The approach used in this research is that the GAN model, which is similar to deep learning models that detect and classify objects in images, is trained to detect the flare artifacts in the flare-degraded images and remove them. The multi-layered framework of the GAN struc-

ture enables it to acquire the complicated characters not only in the flare artifacts but also in the clean images' parts, thus it can solve the original problem and get rid of those unwanted distortions.

2.3.1 Inference Time in Deep Learning

Inference time denotes the duration a trained deep learning model requires to process an input and generate an output. Unlike the training phase, which involves iterative adjustments to model parameters, inference is a forward-only process where the model applies its learned knowledge to new data [13].

Efficiency during inference is highly necessary for the execution of machine learning models in real-world scenarios that require real-time or short response times, such as voice assistants, recommendation systems, or autonomous cars. Inference-time compute, if high, can cause latency to be higher, operating costs to rise, and energy consumption to increase all of which can result in a worse user experience and scalability problems.

Several factors influence the inference time. The model complexity is one of the leading factors; more extensive models with many parameters typically need more resources, and thus have longer running time of inference. The use of hardware acceleration is the other essential point; therefore, the employment of such special hardware as GPUs, TPUs, or custom inference chips greatly accelerates the process. Quantization, i.e., the process of lowering the precision of model weights and activations (for instance, from 32-bit to 8-bit), can decrease computational load with minimal impact on accuracy. Batching, or running many inputs simultaneously, might be a way to increase throughput, but may increase latency for individual predictions. Last but not least, the model architecture itself determines the speed of inference. Some architectures are especially designed to be lightweight and efficient; thus they are suitable for devices with a limited computational power.

Inference time optimization involves strategies such as model distillation, edge deployment, and dynamic models that adjust their complexity based on the input. These approaches aim to balance performance with resource efficiency, ensuring that AI applications will be more responsive and economical.

2.4 Generative Adversarial Networks

Generative Adversarial Networks (GANs), introduced by Goodfellow et al. [14], have revolutionized the field of deep learning by providing a novel approach to data generation. A GAN consists of two neural networks, the **generator** and the **discriminator**, which are trained against each other. The generator creates synthetic data (for example, an image) from random noise, and the discriminator distinguishes between real and generated data. With time, the generator gets better at creating data resembling the original, and the discriminator learns how to discern between original and fake data.

GANs are known to be very successful for many tasks, such as image generation, super-resolution, and image-to-image translation. In particular, for restoration of images, GANs have proven to be highly effective in learning the distribution of clean images from corrupted ones to restore missing or damaged content. Using adversarial loss, GANs really push the network into generating images that are not only accurate but are also very visually pleasing and further preserve the details and textures [15].

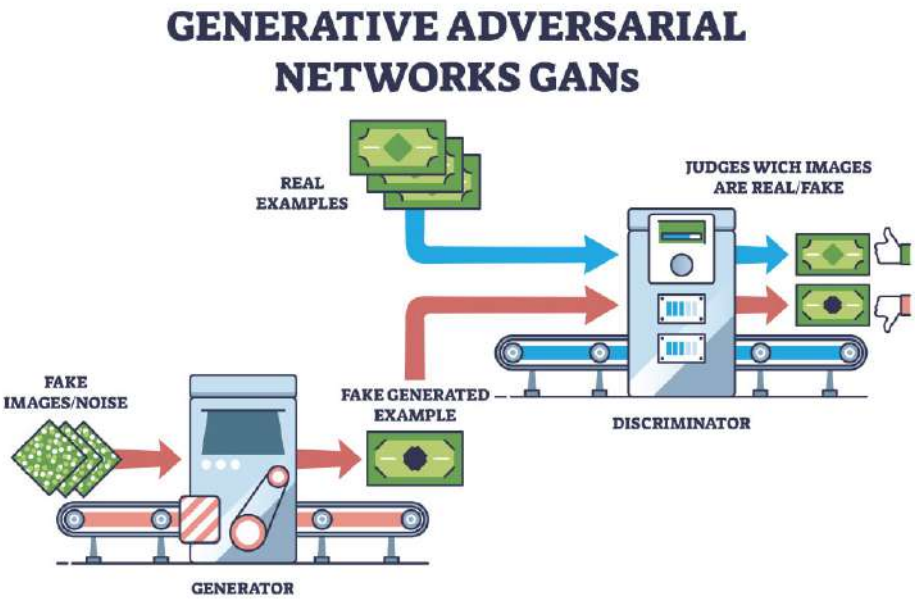


Figure 2.1: An example of image restoration using GANs, adapted from [2].

2.5 Spatial Attention Mechanisms in GANs

Spatial attention mechanisms have become an essential enhancement in modern GANs, especially for tasks that require high-fidelity image generation and accurate detection of manipulated content. These mechanisms draw inspiration from human visual attention, where certain parts of a scene naturally attract more focus based on their relevance or detail [16]. Similarly, in GANs, spatial attention allows the network to identify and prioritize important regions within an image, directing computational resources toward those areas.

Instead of processing the entire image uniformly, the spatial attention module generates an attention map that highlights which parts of the image contribute more significantly to the task at hand—whether it's generating textures or detecting artifacts. This leads to more refined outputs from the generator and more discerning evaluations by the discriminator [17].

For example, in fake image detection, regions like facial features or textured surfaces are often difficult to replicate accurately. Spatial attention helps the discriminator focus on such complex regions, making it easier to identify subtle irregularities introduced during synthesis [18]. On the generation side, models such as SpaGAN (Spatial Attention GAN) have shown that by attending to key areas during image creation, the visual coherence and realism of the output significantly improve [19].

Ultimately, spatial attention mechanisms enhance the ability of GANs to model intricate visual details and support more reliable discrimination between authentic and synthetic images. This is particularly valuable in fields such as digital forensics, content moderation, and media verification, where the integrity of visual data is paramount.

2.6 Metrics PSNR and SSIM

In image processing and computer vision tasks—particularly in image restoration, enhancement, and generation—evaluating the quality of the output is critical. Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) are among the most preferred metrics. These two metrics complement each other: PSNR is concerned with pixel-level accuracy whereas SSIM deals with perceptual and structural similarity, which are frequently more consistent with human visual comparison.

2.6.1 Peak Signal-to-Noise Ratio (PSNR)

Peak Signal-to-Noise Ratio (PSNR) is an image processing objective measure to specify the quality of a reconstruct or compressed image related to its original version [20]. It calculates the ratio between the highest power that a signal can have and the power of the noise, which changes its representation. The PSNR of signals derived from images has a large dynamic range; hence, PSNR is given on a logarithmic decibel (dB) scale.

The PSNR is calculated using the Mean Squared Error (MSE) between the original image I and the processed image K, both of dimensions $m \times n$:

$$\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{{\text{MAX}_I}^2}{\text{MSE}} \right)$$

Here, MAX_I represents the maximum possible pixel value of the image (e.g., 255 for 8-bit images).

Higher PSNR values indicate that the reconstructed image is closer to the original, implying better quality. PSNR is broadly applied in various image processing areas such as compression, denoising, and enhancement to provide an objective input of image quality. Yet, one has to bear in mind that the PSNR might be good for detecting differences on the pixel level but it may not always be consistent with the perceived visual quality since it does not factor in the human visual perception.

2.6.2 Structural Similarity Index (SSIM)

The Structural Similarity Index (SSIM) is a perceptual metric that is used to assess the similarity between two images [21]. The SSIM approach goes beyond pixel-wise differences to take into account of human visual perception, which means that it takes account of the changes in structural information, luminance, and contrast in comparing to human visual perception.

SSIM computes three comparison measurements between the original and the processed images:

- **Luminance:** Evaluates the difference in brightness between the images.

- **Contrast:** Assesses the difference in contrast or the range of pixel intensities.
- **Structure:** Measures the similarity in patterns or structures within the images.

These elements are then represented by a single SSIM figure that is within a range of -1 to 1, where 1 stands for complete similarity, 0 indicates no similarity, and negative numbers point to difference. The extent of similarity between the images is directly proportional to the SSIM number.

One of the advantages of SSIM over MSE and PSNR is its ability to detect more subtle changes in image quality, such as local changes and textures thus, being more compatible with the subjective human evaluations of image quality.

2.6.3 Comparison Summary

Table 2.1: Comparison between PSNR and SSIM Metrics

Aspect	**PSNR (Peak Signal-to-Noise Ratio)**	**SSIM (Structural Similarity Index)**
Definition	Measures the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation.	Assesses the similarity between two images based on luminance, contrast, and structural information.
Calculation Basis	Based on Mean Squared Error (MSE) between the original and processed images.	Combines comparisons of luminance, contrast, and structure between images.
Output Range	Typically 0 to ∞ (expressed in decibels).	-1 to 1, where 1 indicates perfect similarity.
Interpretation	Higher values indicate better image quality; values above 30 dB are generally considered acceptable.	Values closer to 1 indicate higher structural similarity and better perceived image quality.
Sensitivity	Sensitive to pixel-level differences; may not align well with human visual perception.	Sensitive to structural and perceptual differences; aligns more closely with human visual perception.
Computational Complexity	Relatively low; simple to compute.	Higher; involves more complex calculations.
Use Cases	Commonly used for evaluating compression and transmission quality.	Preferred for assessing perceptual image quality in restoration and enhancement tasks.

2.7 Green-Red Vegetation Index (GRVI)

The Green-Red Vegetation Index (GRVI) is a spectral vegetation index that measures the vegetation cover by determining the ratio of green to red reflectance in the visible spectrum. In contrast to traditional vegetation indices, e.g., the Normalized Difference Vegetation Index (NDVI), that uses the near-infrared band, GRVI only utilizes the visible range and thus, it is very suitable in cases when only RGB imagery is available [22].

The GRVI is calculated using the following formula:

$$\text{GRVI} = \frac{\rho_{\text{green}} - \rho_{\text{red}}}{\rho_{\text{green}} + \rho_{\text{red}}}$$

where ρ_{green} and ρ_{red} represent the reflectance values in the green and red bands, respectively.

This index has demonstrated its applicability in monitoring vegetation phenology, particularly in tracking leaf green-up and autumn color changes. It provides high temporal resolution data, which is essential for understanding seasonal variations in vegetation. Correspondingly, the research results have indicated the positive correlation of GRVI with the phenological events recorded on the ground; thus, it is a modest but quite effective instrument in monitoring the movement of vegetation and its good condition.

Because of its capability to reflectance at different parts of the visible spectrum, GRVI is also expected to be a key tool for the improvement of flare removal in remote sensing images. If we remove the flare from an image we have to be sure that the resulting image still accurately imitates natural vegetation and that the process has not given rise to new errors or has not lost important parts. As a result of the green to red reflectance ratio in both the original and the restored images is used, GRVI acts as an unbiased indicator imitating the level of good flare removal while at the same time it makes possible the continuity of the vegetative structures.

In particular, GRVI can help evaluate the restoration quality by ensuring that the restored image maintains a similar green-to-red reflectance ratio as the original, unflared image. This can be particularly useful for detecting areas where the flare removal process may have caused distortions or misrepresentations of the vegetation cover. If the restored images have higher GRVI numbers than the ones which contain the flare, the restoration will be considered successful, preserving the true vegetation characteristics.

In summary, GRVI is not only a valuable tool for remote sensing vegetation studies but

also serves as a robust metric for assessing the quality of flare-free images, ensuring that flare removal techniques achieve both high fidelity and perceptual accuracy in the final restored images.

2.8 Tools and Libraries

The development of the system involved several powerful tools and libraries to handle different aspects of the project, from virtual environment creation to deep learning implementation and real-time processing. The system was built on a robust hardware setup, specifically designed to ensure optimal performance for image restoration tasks and the deep learning model training. The hardware configuration consisted of an NVIDIA GPU with CUDA support, a multi-core CPU, 32 GB of RAM, and high-speed SSD storage, which together provided the necessary computational power for handling large-scale datasets and model training efficiently.

2.8.1 Hardware Setup

The hardware setup for training and evaluating the image restoration model was designed to ensure optimal performance and efficiency. The system utilized the following components:

- **GPU**: NVIDIA A40 with CUDA 12.4 support was used to accelerate the training and inference processes. The GPU enabled parallel computing, significantly reducing the processing time for training the models involved in the flare removal task.
- **CPU**: The configuration included two Intel Xeon Gold 6330 CPUs, each offering 28 cores and 56 threads, totaling 56 physical cores and 112 threads, for general-purpose tasks. However, most of the computational load was offloaded to the GPU, ensuring that the system could handle large neural network models efficiently.
- **Memory**: The system was equipped with 192 GB of RAM, ensuring smooth data handling during training and inference. This memory configuration supported the large batch sizes required during model training.
- **Storage**: High-speed SSD storage was used to store the dataset and the generated models. This configuration allowed for rapid data loading and saving of intermediate checkpoints during the training process.

This hardware setup provided the necessary computational resources to efficiently train and evaluate the image restoration model, ensuring that the system could handle the demanding nature of the task.

2.8.2 Unreal Engine

Unreal Engine is a comprehensive software package and widely known game engine created by Epic Games, mostly recognized for its excellent rendering abilities and real-time simulation. It provides a suite of tools designed for game development, film production, architecture, and simulation. The platform offers a high-performance rendering pipeline, making it perfect for producing photorealistic scenes with real-time rendering [23].

Unreal Engine's support for both Blueprint Visual Scripting and C++ programming allows developers to create complex interactive applications without deep programming knowledge, while still offering flexibility for more advanced users. The engine also comes with such features as AI development, multiplayer networking, and VR/AR support, which fits the nature of the work in this project.

In the context of this thesis, Unreal Engine was used to solve the flare removal task by generating synthetic data, thus allowing the simulation of environmental conditions and flare effects of various kinds. Its real-time rendering was the main feature that allowed one to create a dynamic and visually realistic wheat field, which was the basis for deep learning model training.

2.8.3 PyTorch

PyTorch is an open-source machine learning framework developed by Meta and is extensively used for deep learning purposes. It is based on the Torch library, which was originally based on the Lua scripting language; however, PyTorch has provided a more user-friendly Python interface, making it the most popular framework in the machine learning society [24].

The most prominent feature of PyTorch is that it works with dynamic computation graphs, also known as define-by-run graphs. In other words, the computational graph structure is dynamically formed during runtime, giving more flexibility than the static graph frameworks. The feature is particularly great for research and experimentation, where the model structure can undergo frequent changes.

Moreover, PyTorch goes with a tensor-based method, with its `torch.Tensor` class giving a structure that supports multi-dimensional arrays and GPU acceleration which makes it similar to the array-like one. This results in fast computations, which is integral when training deep neural networks on large datasets. Moreover, PyTorch makes use of `autograd` module to provide the automatic differentiation that facilitates the efficient calculation of gradients which is important in optimizing neural networks during training.

In this thesis, the model used for flare removal was developed using PyTorch. PyTorch was chosen for its flexibility, ease of use, and comprehensive support for building and training deep neural networks. Its dynamic computation graphs and seamless integration with GPU hardware facilitated efficient model training, which was essential for handling large-scale image restoration tasks where both performance and efficiency are critical.

2.8.4 OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library which provides a full set of tools for real-time image and video processing. OpenCV was originally created by Intel and is now one of the most frequently used libraries in the research community as well as in the industry. OpenCV enables the user to perform a variety of tasks, such as facial recognition, motion detection, image segmentation, and real-time object tracking.

In this thesis, OpenCV was used for real-time implementation, where it played a key role in handling tasks such as capturing images, processing them, and displaying the results. The ability of OpenCV to efficiently manage large image data and perform image processing operations, including filtering, transformation, and feature extraction, made it the indispensable tool in the model's workflow. Its extensive compatibility with numerous image types and, on top of that, its capability to work alongside deep learning models facilitated the bridging of the gap between deep learning frameworks (PyTorch) and real-time image processing tasks, for instance, flare removal in videos and images.

Chapter 3

Related Work

Early efforts in flare artifact removal were largely limited by the lack of high-quality training data. Wu et al. [10] introduced the first general-purpose, learning-based approach to remove lens flare from a single image. The method of the authors tackles the big challenge of obtaining paired flare-affected and clean image data by suggesting a physics-based data synthesis pipeline. This pipeline is a mix of the simulated scattering flare done with wave optics and the reflective flare that is empirically captured, which makes it possible to train deep neural networks, namely U-Net and context aggregation networks, on a semi-synthetic dataset. To make sure that no light sources are removed unintentionally during training, they came up with a new loss function that masks the saturated regions and hence the light sources can still be there while training. The trained models showed that they can generalize a lot of different scenes, cameras, and flare types, and they further boosted the performance of the downstream vision tasks such as semantic segmentation and monocular depth estimation.

Complementary to this direction, Fu et al. [3] tackled the problem of image degradation from non-homogeneous haze using a frequency-aware approach. They came up with DW-GAN, which integrates discrete wavelet transforms (DWT) into a dual-branch GAN framework. One branch concentrates on grasping low-frequency structural parts, whereas the other branch polishes high-frequency details. The authors also propose a knowledge adaptation module by employing Res2Net features to fill the gap between the two domains, synthetic and real. Their method achieves state-of-the-art performance on the NH-Haze benchmark, showcasing the efficacy of frequency-domain modeling—a concept parallel to spatial attention in improving localized artifact removal, as explored in this thesis.

Attention mechanisms in GANs have also shown promise in tasks requiring spatially pre-

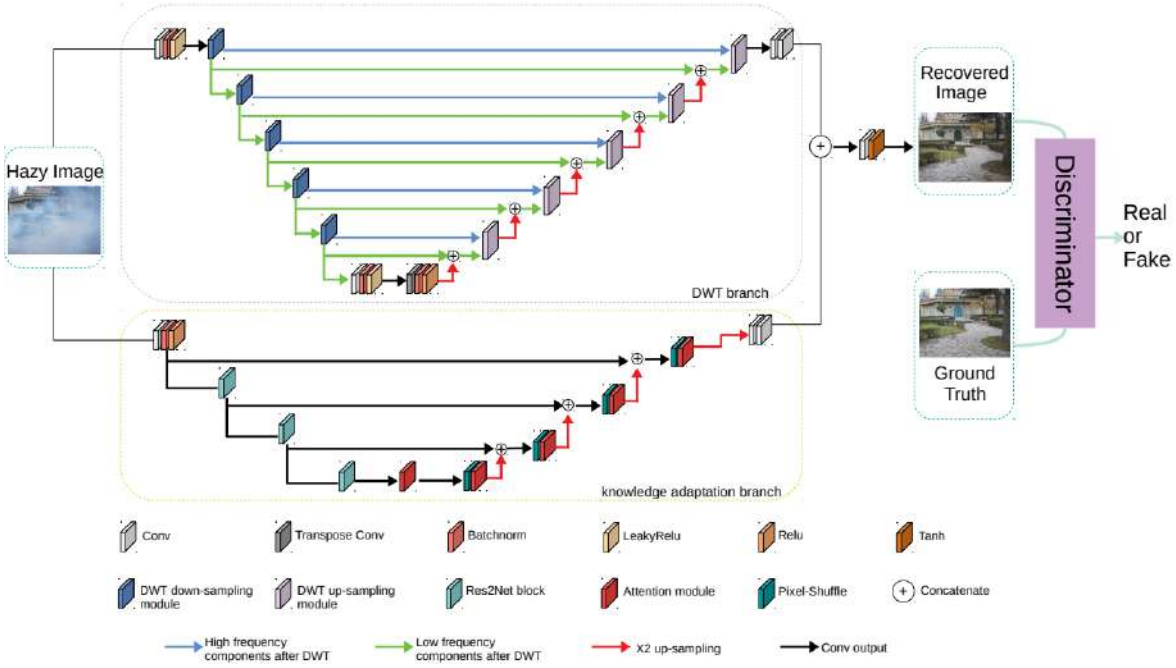


Figure 3.1: The network structure of the proposed GAN [3].

cise restoration. A notable example is the work of Qian et al. [4], who proposed the Attentive GAN to remove raindrops from images. Their strategy is based on the use of an attention map in the generator and the discriminator as a localization tool, which allows the recovery of raindrop-degraded regions. The dual-attention concept thus allows the generator to create the parts of images which are invisible due to occlusions, and the discriminator to become a detector of such parts. In addition to the adversarial and perceptual losses, they showed qualitative and quantitative evidence to support the claim that their method achieves state-of-the-art performance in multiple datasets and thus provides a basis for various spatially guided/restoration techniques that can be efficiently applied to flare removal.

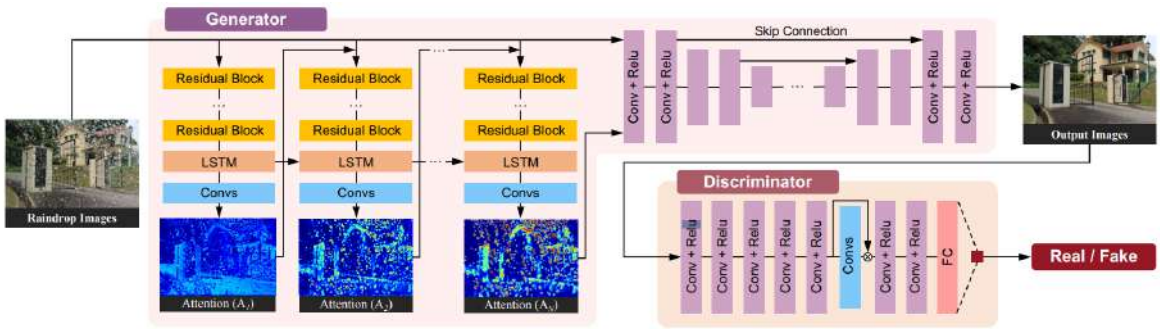


Figure 3.2: The architecture of proposed attentive GAN [4].

Based on this trend, Pan [5] introduced SpA-GAN, a spatial attention-based GAN cre-

ated specifically for the task of removing thin and thick clouds from satellite images. The design increases the ability of the GAN when it plugs in a spatial attention module, which not only locally magnifies the cloud-affected regions, but also maintains global consistency. This attention mechanism dynamically adjusts the network's focus during training, allowing it to recover scene details otherwise obscured by irregular, semi-transparent artifacts. Quantitative evaluations reveal significant improvements in PSNR and SSIM compared to baseline methods. In addition, the novel attention-driven design and the suitability for the visual noise phenomenon of the SpA-GAN model made it the starting point of this thesis' work on flare removal.

Chapter 4

Implementation: Synthetic Dataset Generation in Unreal Engine

4.1 Environment Configuration and Dataset Synthesis

In order to train and evaluate the proposed model for flare artifact removal, a custom synthetic dataset was generated using Unreal Engine 5. The objective was to achieve a controlled, yet visually realistic virtual environment that could simulate natural outdoor scenes in which lens flares may occur. Unreal Engine was thus selected because of its superior rendering capabilities as well as its ability to simulate complex lighting, materials, and camera effects.

The virtual scene was created to be like a wheat field, with a number of assets and textures to maintain a high level of visual fidelity. In addition to environment modeling, material configuration, lighting setup, and lens flare simulation were implemented in such a way as to add variety and realism to the dataset. Thus, conditions encompassing lighting and camera angles and all sorts of flare cases form an image set that serves as a very good foundation for supervised training.

4.1.1 Unreal Engine Setup

Unreal Engine 5 was chosen for its robust capabilities in creating detailed and dynamic environments, especially for generating synthetic datasets. The virtual environment of the wheat field that was built could be subjected to changes in lighting and camera angles to trigger lens flare systematically. Unreal Engine's real-time rendering engine ensured the ex-

istence of environmental effects that would realistically foster the generation of flare artifacts in images.

4.1.2 Materials and Textures

To enhance the realism of the scene, a wide variety of materials and textures were applied to objects within the environment. These included textures of soil, rocks, mountains, grasses, chilling and dry wheat, trees, hay bales, electricity poles, houses, and warehouses. All assets were obtained from Fab [25], which is an internet library of high-quality 3D materials that extend support to the Unreal Engine.

Figure 4.1 shows a selection of these materials as applied to the environment. As illustrated, textures were selected and mapped in the landscape and objects to simulate typical ground surfaces and vegetation found in wheat fields. Wheat crops required great attention to detail; green and dry variants, as well as different planting densities and heights, were modeled. This increased the visual variance of the scene and attempted to better reflect the conditions encountered in actual agricultural environments.

(i) Green wheat [26]

(ii) Dry wheat [26]

(iii) Soil, grass and hay ball [27] [28] [29]

(iv) Electricity Column [30]

(v) Trees [31]

(vi) Mountains [32]

(vii) House

(viii) Warehouse [33]

Figure 4.1: Overview of materials and textures used in the virtual environment, showcasing various ground surfaces, vegetation types, and structural elements.

These material configurations contributed significantly to the photorealism of the syn-

thetic dataset and played a crucial role in ensuring that the generated images closely resembled real outdoor agricultural scenes.

Figure 4.2: Scene with dry wheat materials simulating ready-to-harvest field.

Figure 4.3: Scene with green wheat materials simulating early growth field.

4.1.3 Lighting Simulation

A key component of the dataset generation pipeline was the use of Unreal Engine's directional light to replicate realistic sunlight conditions. Adjustments to the directional light were made according to different times of day-such as morning, noon, or afternoon-rotation and

intensity. As a result, dynamic shadowing occurred alongside changes in scene illumination, slightly increasing the variety of visuals available in the generated dataset.

While Unreal Engine allows real-time manual adjustment of the sun direction using the shortcut `Ctrl+L`, for consistency and reproducibility across the dataset, the lighting simulation in this work was driven by a structured combination of predefined pitch and yaw values. Specifically, two vertical angles (pitch) and three horizontal angles (yaw) were selected to cover a representative range of sunlight positions. These combined allow simulating various lighting directions and intensities that would be cast by the sun corresponding to different heights and azimuths throughout the day.

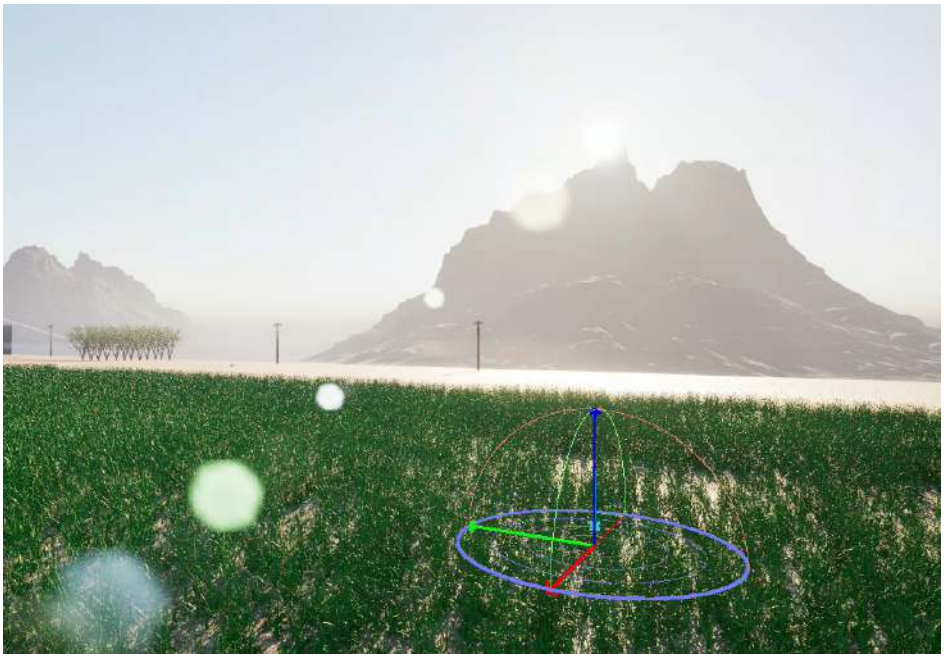


Figure 4.4: Directional light setup in Unreal Engine used for lighting simulation.

Additionally, cloud layers were integrated into the environment to simulate light scattering and occlusion effects, further enriching the lighting conditions with more natural variation and realism.

Figure 4.5: Volumetric cloud setup in Unreal Engine used to simulate realistic lighting conditions.

4.1.4 Lens Flare Simulation

To replicate real-world optical artifacts observed in camera systems, the PostProcessVolume in Unreal Engine was utilized to simulate lens flares. Specifically, the ”Lens Flares” section of the PostProcessVolume was enabled, and the following parameters were adjusted:

- **Intensity:** Set to 1.0 for all variations.
- **Bokeh Size:** Varied across four distinct settings—no adjustment (default), 5.0, 10.0, and 16.0.

Using these settings, four unique flare cases were created:

1. Flare1: Flare with **intensity 1.0** and default bokeh size (see Figure 4.6).
2. Flare2: Flare with **intensity 1.0** and **bokeh size 10.0** (see Figure 4.7).
3. Flare3: Flare with **intensity 1.0** and **bokeh size 16.0** (see Figure 4.8).
4. Flare4: Flare with **intensity 1.0** and **bokeh size 5.0** (see Figure 4.9).

Figure 4.6: Flare1.

Figure 4.7: Flare2.

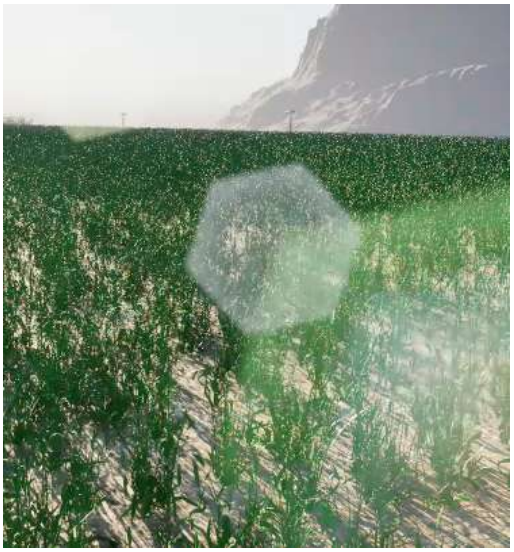


Figure 4.8: Flare3.

Figure 4.9: Flare4.

These variations allowed the dataset to include a diverse range of lens flare artifacts in terms of shape and spread, helping the model generalize better in learning how to identify and remove such artifacts. The use of PostProcessVolume provided full control over the optical simulation, ensuring consistency and realism in the synthetic data.

4.1.5 Waypoint-Based Navigation Paths

To guide the AI player through the virtual wheat field and capture diverse views under varying environmental conditions, a network of waypoints was strategically placed across

the map. These waypoints acted as navigational markers, defining distinct paths for the AI to follow in the virtual environment. Each path was carefully designed to simulate realistic traversal patterns, such as moving forward, backward, or diagonally across the field.

The paths used during dataset generation are outlined below, where each number corresponds to a specific waypoint (e.g., 1 refers to `Waypoint1`), and the direction denotes the general orientation of the AI player's movement along that path:

Path ID	Waypoints	Orientation
Path1	1–12	Left Forward
Path2	13–24	Left-Middle Forward
Path3	1, 14, 25–34	Forward
Path4	35–46	Right
Path5	1, 48–58	Left Backwards
Path6	59–70	Right Backwards
Path7	70–59	Left (Reverse Traversal of Path6)

Table 4.1: Defined navigation paths using AI-controlled waypoints in the virtual wheat field.

These paths were executed using Unreal Engine's AI pathfinding system in conjunction with the blueprint-controlled AI Player. A waypoint array was used to hold the ordered sequence of waypoints corresponding to each selected path, and a navigation mesh (NavMesh) was created to specify the walkable area within the environment, thus allowing smooth and realistic movement of the AI character along the predefined routes. The adoption of structured waypoint sequences guaranteed that the camera movement patterns were both consistent and repeatable, thereby facilitating the collection of training images with well-distributed spatial coverage and a wide range of perspectives. This diversity is essential for the improvement of the generalizability of the GAN model, as it allows the network to be familiar with different backgrounds, object scales, and illumination conditions throughout the scene.

4.1.6 Parameterization of Dataset Cases

To ensure high variability and robustness in the training dataset, a total of 20 unique cases were generated using Unreal Engine, each capturing a distinct combination of environmental and camera-related parameters. These cases allowed the simulation of real-world conditions

due to changes in parameters such as flare types, light source angles, terrain textures, navigation paths, and weather conditions. Each case produced a substantial number of paired images—ranging from 49 to 144 pairs—resulting in a rich and diverse dataset. The cases are systematically named and characterized using a structured notation format that reflects the key parameters used during their creation.

- **Flare**: The type of lens flare applied in the case (e.g Flare1, Flare2, etc.)
- **Path**: Refers to the navigation route taken by the AI character, defined by a sequence of Unreal Engine waypoints (e.g., Path1, Path2, etc.).
- **Pitch/Yaw Angles**: Indicate the sun's vertical (pitch) and horizontal (yaw) positions, used to simulate different times of day and illumination angles via the directional light component in Unreal Engine.
- **X Shift**: The horizontal displacement of the camera path in the environment (e.g., `X+900`, `X-500`) to diversify capture locations across the scene.
- **Δ Waypoints**: Specifies the distance between consecutive waypoints along the path, affecting the movement granularity and scene sampling resolution.
- **Weather**:
  - **NC**: No Clouds (clear sky)
  - **C**: Cloudy weather (volumetric clouds enabled)
- **Grass Type**:
  - **D**: Dry grass
  - **G**: Green grass
- **Direction (Dir.)**: Denotes the overall movement orientation of the AI camera:
  - **LF**: Left Forward
  - **F**: Forward
  - **R**: Right
  - **RB**: Right Backwards

– **LB**: Left Backwards

This naming convention and the resulting case diversity enable the dataset to cover a wide range of real-world conditions, supporting the training of a robust GAN model for lens flare removal in agricultural environments.

Case	Image Pairs	Flare	Path	Pitch	Yaw	X Shift	Δ Waypoints	Weather	Grass	Dir.
1	144	Flare1	Path1	-9/-12	145/155/165	+900	1000	NC	D	LF
2	144	Flare2	Path2	-9/-12	145/155/165	+900	1000	NC	D	LF
3	102	Flare3	Path3	-7/-10	165/175/185	+900	1200	NC	D	F
4	102	Flare1	Path3	-7/-10	165/175/185	+900	1200	NC	D	F
5	102	Flare1	Path4	-6/-9	220/230/240	+900	1200	NC	D	R
6	102	Flare2	Path4	-6/-9	220/230/240	+900	1200	NC	D	R
7	96	Flare4	Path5	-5/-8	315/325/335	-500	700	NC	D	LB
8	96	Flare3	Path5	-5/-8	315/325/335	-500	700	NC	D	LB
9	90	Flare1	Path6	-6/-9	400/410/420	-500	700	NC	G	RB
10	96	Flare2	Path6	-8/-12	420/440/460	-500	700	NC	G	RB
11	90	Flare2	Path7	-8/-12	240/260/280	+500	700	NC	G	R
12	90	Flare1	Path7	-8/-12	240/260/280	+500	700	C	G	R
13	126	Flare3	Path2	-8/-12	160/170/180	+900	1000	C	G	LF
14	84	Flare1	Path4	-8/-12	220/240/260	+900	1200	C	G	R
15	81	Flare1	Path3	-1/-3	165/175/185	+900	1200	C	G	F
16	90	Flare1	Path3	-5/-8	160/170/180	+800	1200	C	G	F
17	49	Flare1	Path2	-7/-10	120/130/140	+700	1000	NC	G	LF
18	120	Flare1	Path2	-8/-11	155/175/195	+700	1000	NC	G	LF
19	72	Flare1	Path4	-5/-8	215/225/235	+700	1200	C	G	R
20	126	Flare2	Path2	-8/-11	155/175/195	+700	1000	NC	G	LF

Table 4.2: Full set of dataset generation cases with corresponding environmental and path parameters.

The following figures present one sample image from each of the 20 scenarios, highlighting the diversity of visual conditions introduced during dataset generation.

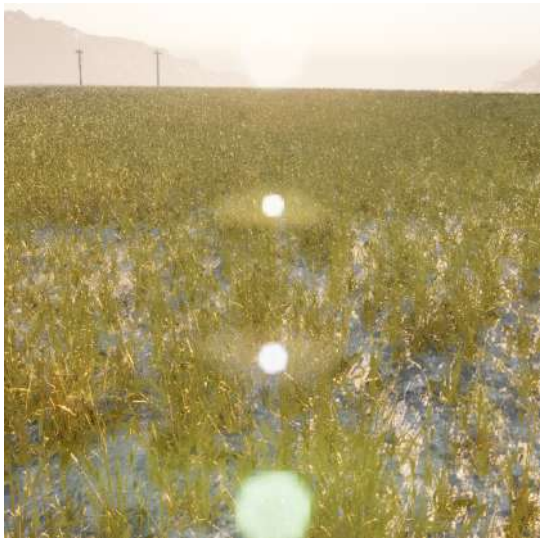


Figure 4.10: Case 1.

Figure 4.11: Case 2.

Figure 4.12: Case 3.

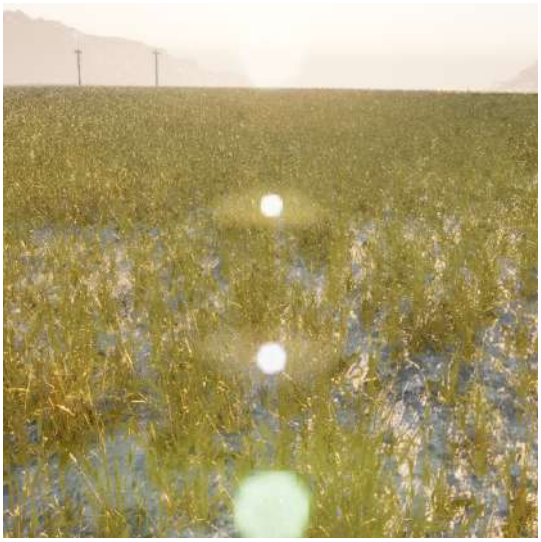


Figure 4.13: Case 4.

Figure 4.14: Case 5.

Figure 4.15: Case 6.

Figure 4.16: Case 7.

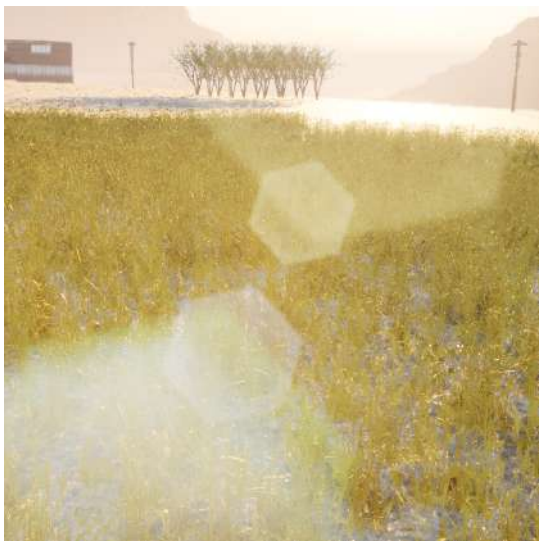


Figure 4.17: Case 8.

Figure 4.18: Case 9.

Figure 4.19: Case 10.

Figure 4.20: Case 11.

Figure 4.21: Case 12.

Figure 4.22: Case 13.

Figure 4.23: Case 14.

Figure 4.24: Case 15.

Figure 4.25: Case 16.

Figure 4.26: Case 17.

Figure 4.27: Case 18.

Figure 4.28: Case 19.

Figure 4.29: Case 20.

4.2 Contents of the Custom Flare Dataset

During the development process, multiple versions of the training dataset were generated and evaluated in order to maximize model performance and ensure robustness across diverse flare scenarios. Each version was curated based on the presence and quality of flare artifacts, as well as the variation in environmental conditions and paths. The following summarizes the key characteristics of each dataset:

Dataset 1: This initial dataset included a total of **849 paired images**, primarily featuring flare type **Flare1 (F1)**, along with a smaller number of samples from **Flare2 (F2)** and **Flare4 (F4)**. It was composed of image pairs from the following cases: **1, 2, 4, 5, 7, 12, 15, 16**.

Dataset 2: It was observed that **Dataset 1** contained a significant number of images without visible flare, which degraded training quality. To address this, a refined version was constructed by selecting only the samples with prominent flare artifacts from Dataset 1, resulting in **622 paired images**. This version also included image pairs from additional flare-heavy cases: **17, 18, 19, 20**. The total size of this curated dataset was **989 image pairs**.

Dataset 2.5: This dataset was independently constructed using newly collected image pairs not included in previous datasets. It consists of **537 paired images** selected from cases **3, 6, 8, 9, 10, and 11**, all characterized by prominent flare artifacts and varied scene compositions. The goal was to introduce fresh flare patterns and scene diversity into the training process. This dataset later served as a key component in the formation of **Dataset 3**.

Dataset 3: This dataset was formed by combining the entirety of **Dataset 2** (989 image pairs) and **Dataset 2.5** (537 image pairs), resulting in a total of **1,526 paired images**. By incorporating diverse and flare-rich samples from both curated and newly collected cases, this version represents the most comprehensive and flare-focused dataset used during training, enhancing the model's ability to generalize across various flare types and scene conditions.

These progressive refinements allowed the model to be trained on a diverse and representative set of flare-affected agricultural scenes, improving its generalization capacity and real-world applicability.

4.3 Blueprints and Code for Environment Setup

In order to assist the automated dataset generation and environmental control a union of the Unreal Engine blueprints and C++ scripts was created. Blueprints were in charge of managing the logic on a scene level such as the placing of dynamic objects and environmental state transitions. On the other hand, C++ code was utilized to regulate the directional light (which serves as the sun), thus, simulating the solar movement that is realistic by going

through six predefined combinations of pitch and yaw angles. This allowed the creation of the lighting conditions, which were different from each other, spread all over the environment.

To provide paired examples under the same conditions for every position of the sun, two pictures were taken, first with the lens flare on and the other with it off. Then every image was cropped from the center area only to a size of 512×512 pixels. This step has been experimentally proven to be the most efficient in highlighting not only the wheat field, but also the flare effects that come along with it and still being efficient for training purposes.

4.4 AI Player for Camera Movement and Image Capture

An AI Player was created using Unreal Engine's blueprint system to automate camera movement through the virtual wheat field. The AI character was programmed to follow a path defined by a network of waypoints, enabling it to traverse the environment in a natural and continuous manner. This movement ensured that images were captured from diverse locations and perspectives, with the camera positioned at a height and angle representative of a tractor-mounted system commonly used in precision agriculture.

The AI Player was set to move at a speed of **190 cm/s**, which is approximately **6.84 km/h**. This speed was selected to represent the average moving speed of agricultural tractors during field monitoring, ensuring realistic temporal sampling and smooth dataset generation.

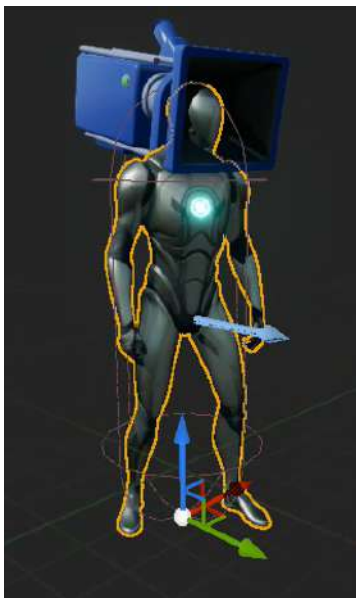


Figure 4.30: AI Player.

The AI also synchronized with the C++ capture logic to activate the image capture at

certain points and directions. By merging automated navigation with consistent lighting, this method opened a door for the creation of a diverse and representative dataset, which is of great importance in training a GAN that can cope with the complicated lens flare artifacts in the images of the agricultural sector.

Chapter 5

Implementation: SpA-GAN Training and Real-Time System Design

5.1 SpA-GAN Architecture

Spatial Attention Generative Adversarial Network (SpA-GAN) is a GAN architecture designed for the specific purpose of thin-cloud removal in high-resolution remote sensing images of the Earth [5]. It combines the power of generative adversarial training with spatial attention mechanisms that allow the network to selectively focus on cloud-covered regions during both learning and inference. The architecture describes two primary parts, namely the generator and the discriminator, with a composite loss function that incorporates attention supervision.

5.1.1 Generator: Spatial Attentive Network (SPANet)

The generator is based on a convolutional architecture known as the Spatial Attentive Network (SPANet). Its structure is designed to progressively identify and correct cloud-covered areas through a sequence of residual and attention-driven blocks.

- **Initial Feature Extraction:** The input image is first passed through a standard convolutional layer to extract low-level features.
- **Residual Encoding:** This is followed by three standard residual blocks, which help to stabilize learning and preserve context through skip connections.

- **Spatial Attention Processing:** The feature map is then processed through four Spatial Attentive Blocks (SABs). Each SAB is composed of three Spatial Attentive Residual Blocks (SARBs) and one Spatial Attention Module (SAM) in parallel.
- **Final Reconstruction:** After the attention-guided refinement of the features, the output is passed through two more residual blocks and a final convolutional layer to produce the cloud-free image.

The Spatial Attention Module (SAM) utilizes a two-stage, four-directional recurrent neural network (IRNN) to gather local and global contextual information. This allows the network to generate precise attention maps that highlight cloud regions, guiding the SARBs to perform a localized correction.

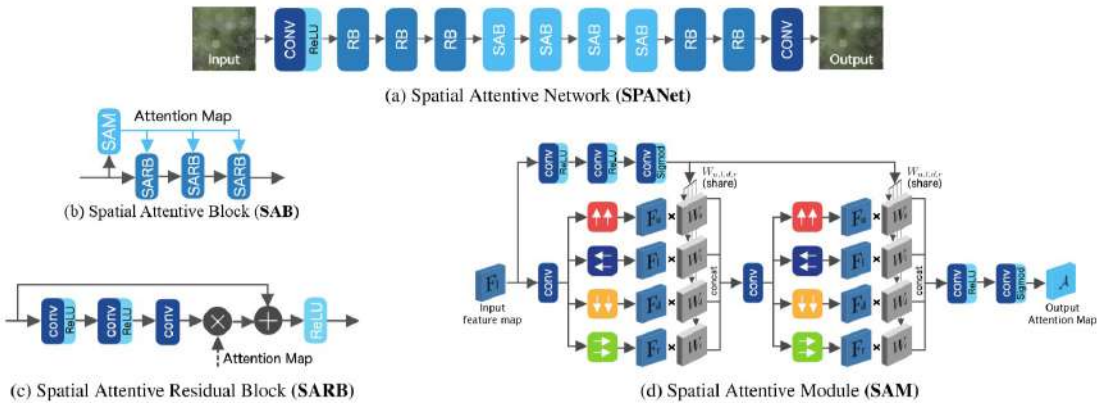


Figure 5.1: SpA-GAN Generator Architecture. (a) The generator (SPANet) includes residual blocks (RB) and spatial attentive blocks (SAB). (b) Each SAB combines attention maps from the Spatial Attention Module (SAM) with Spatial Attentive Residual Blocks (SARB). (c) A SARB applies attention-guided residual learning. (d) SAM uses directional IRNNs to generate the attention map [5].

5.1.2 Discriminator

The discriminator is basically a convolutional neural network that decides whether images are real (ground-truth cloud-free) or fake (generated by the generator). It is composed of a series of convolutional layers with batch normalization and LeakyReLU activation. In this process, the component serves as the source of adversarial training pressure, driving the generator to generate outputs that are indistinguishable from the real cloudless images.

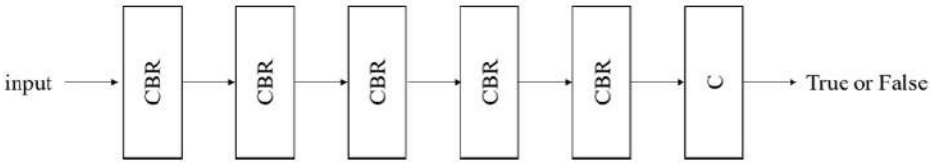


Figure 5.2: SpA-GAN Discriminator Architecture [5].

5.1.3 Loss Function

The training process is guided by a composite loss function:

- **Adversarial Loss (L_{cGAN}):** Encourages the generator to produce output that fools the discriminator.

$$L_{cGAN}(G,D) = E_{x,y\sim p_{data}(x,y)}[\log D(x,y)] + E_{x\sim p_{data}(x),z\sim p_z(z)}[\log(1-D(x,G(x,z)))] \tag{5.1}$$

- **Pixel-wise L1 Loss (L_{L1}):** Minimizes the absolute difference between the predicted and ground-truth images to ensure structural and color accuracy.

$$L_{L1}(G) = \frac{1}{CHW}\sum_{c=1}^{C}\sum_{v=1}^{H}\sum_{u=1}^{W}\lambda_c \left| I_{gt}^{(u,v,c)} - G(I_{in})^{(u,v,c)} \right|_1 \tag{5.2}$$

- **Attention Loss (L_{Att}):** Guides the generator's spatial focus by comparing the predicted attention map with a binary cloud mask.

$$L_{Att} = \|A - M\|_2^2 \tag{5.3}$$

The total loss is defined as:

$$L_{SpA} = \arg\min_G \max_D L_{cGAN}(G,D) + L_{L1}(G) + L_{Att} \tag{5.4}$$

5.1.4 Analysis of Attention Maps Across SPANet Layers

In the proposed GAN architecture, spatial attention is progressively applied through four successive *Spatial Attentive Modules (SAMs)*. These modules enable the network to focus on regions of the input image that require the most restoration, with particular emphasis on lens flare artifacts.

Figure 5.3: Flared.

Figure 5.4: Restored.

The evolution of attention across these layers reveals how the network incrementally refines its understanding and correction of flare effects:

Attention1 (Early Layer – Initial Detection) The network in this phase is extremely focused on those regions that are most corrupted. The attention map highlights bright flare artifacts, such as localized lens flare blobs, as well as the sun region itself, which emits intense light disrupting surrounding pixels. Additionally, vertical or radial flare streaks extending from the sun receive strong focus. This early layer acts as a coarse detector which aids the network in the rapid localization of the visually dominant distortions with sharply defined and concentrated attention.

Figure 5.5: Flared.

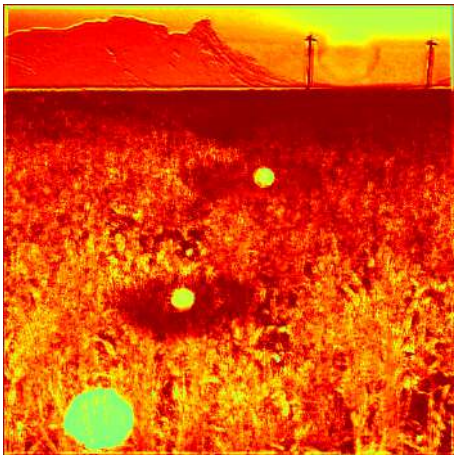


Figure 5.6: Attention Map 1.

Attention2 (Mid Layer – Contextual Expansion) In the second attention map, the network extends its concentration not only on the flare blobs but also on near regions. Greater emphasis is given to the gentle changes that are the result of the scattering of light, especially in the places where the flare and the background are becoming one. The attention becomes more diffuse, reflecting the model's understanding of the flare in the neighboring spatial area rather than the flare as individual parts.

Figure 5.7: Flared.

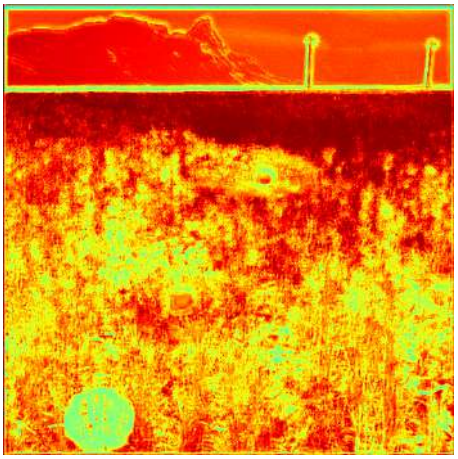


Figure 5.8: Attention Map 2.

Attention3 (Deeper Layer – Semantic Refinement) By this deeper stage, attention is spread over larger, more uniform regions such as grass or terrain affected by global brightness shifts. Although the network's focus still lies on the flare region, it also takes into account the secondary effects that flare may cause such as reduced contrast or color imbalance in the surrounding area. The attention gradually changes from just spatial to semantic one thus deeper network layers get direction to do more detailed corrections which results in better image quality overall.

Figure 5.9: Flared.

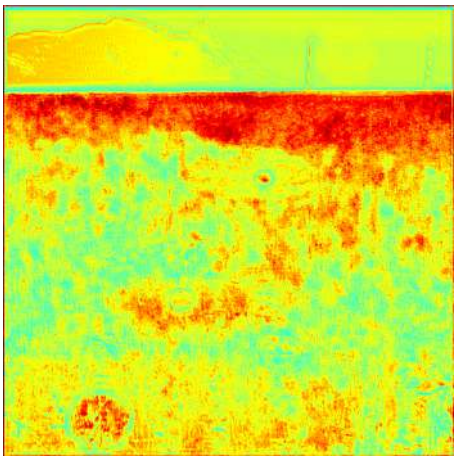


Figure 5.10: Attention Map 3.

Attention4 (Final Layer – Global Balancing) The final attention map displays a smoother and lower-intensity distribution. The network's focus shifts towards ensuring global consistency, making sure the restored image looks natural and free of artifacts. Although flare-affected areas still attract attention, the lower contrast in this map indicates the network's confidence in previous correction steps. This final layer acts as a regulator for the restoration process thus it is able to carry out subtle changes but not to the extent of overcorrecting those parts that are already clean.

Figure 5.11: Flared.

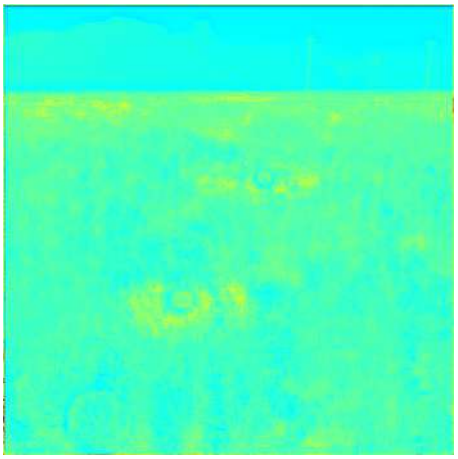


Figure 5.12: Attention Map 4.

5.2 SpA-GAN Model on Cloud Dataset

5.2.1 Testing the Pretrained SpA-GAN Model on Cloud Dataset

Before training the network from scratch, the official pretrained SpA-GAN model provided by the original authors was evaluated on both the RICE1 and RICE2 datasets. This served as a baseline for assessing the effectiveness of custom training configurations. The pretrained model was directly applied to the test subsets of each dataset without any fine-tuning.

Results on RICE1

While testing the RICE1 data with thin clouds, the pretrained model achieved a PSNR of 27.54 dB and an SSIM of 0.872 in the test dataset. Visually, the model demonstrated strong performance in reconstructing fine details and background textures, particularly in areas with sparse cloud coverage. The output images were very close to the ground truth images and did not have many artifacts.

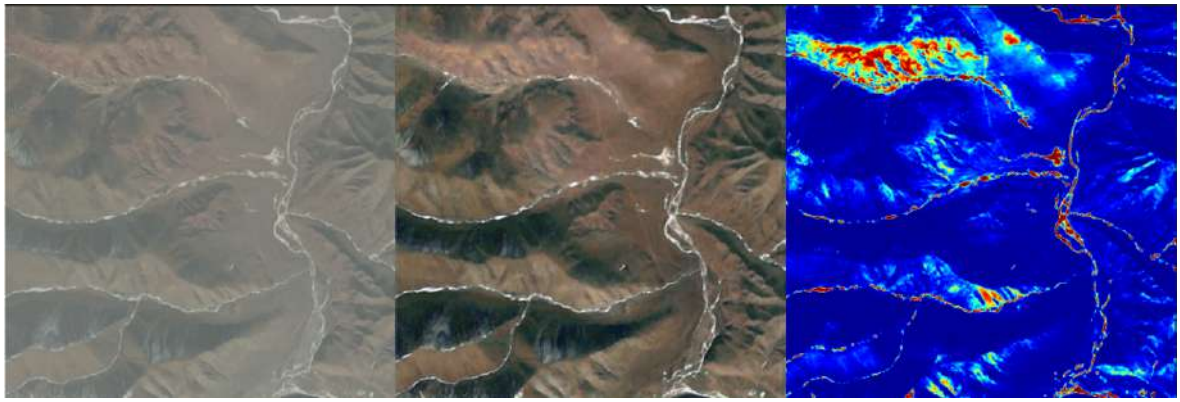


Figure 5.13: RICE1 Result. Left: Flared input image. Center: Restored Image. Right: Attention map showing focus areas during restoration (red/yellow: high attention, blue: low attention).

Results on RICE2

For the more difficult RICE2 dataset, the pretrained model got a PSNR of 25.83 dB and an SSIM of 0.823 in the test dataset. Although the model performed well in semi-occluded regions, thick cloud areas occasionally resulted in blurry reconstructions or texture mismatches.

Nevertheless, the pretrained model provided a solid benchmark, especially considering the complexity of the RICE2 imagery.

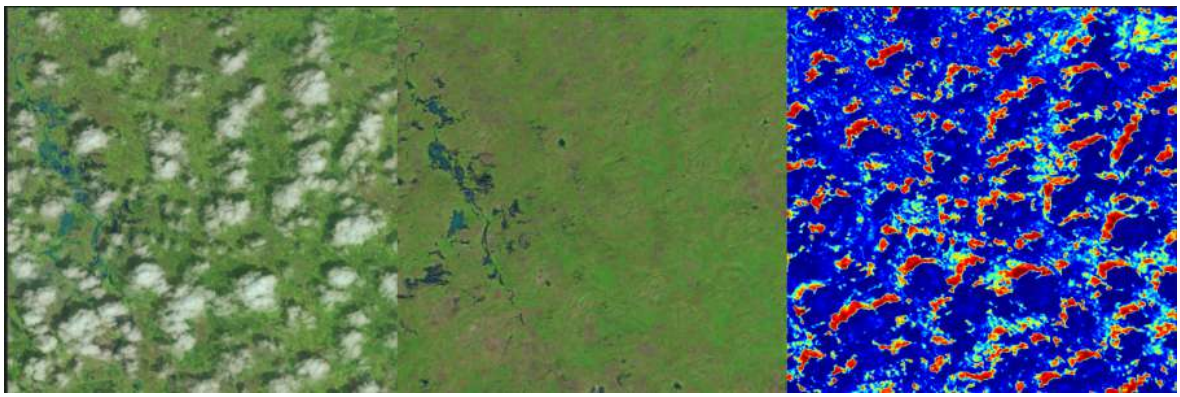


Figure 5.14: RICE2 Result. Left: Flared input image. Center: Restored Image. Right: Attention map showing focus areas during restoration (red/yellow: high attention, blue: low attention).

5.2.2 Preliminary Training on Cloud Dataset

To validate and replicate the performance of the SpA-GAN architecture proposed in the original paper [5], we conducted independent training experiments using the RICE1 and RICE2 datasets. These datasets are composed of satellite images of very high resolution for cloud removal research, and each consists of paired cloudy and cloudless images. The experiments aimed to not only reproduce the pretrained model's results but also to investigate how changing parameters affects the model performance.

Training with RICE1 (Thin Clouds)

The RICE1 dataset includes 500 image pairs (cloudy and cloud-free), each of resolution 512×512 pixels. The images were taken from Google Earth by toggling the cloud layer on and off. The batch size, data split, and learning rate were varied across different training configurations. The baseline was set with a fixed learning rate of 0.0004 and 200 epochs of training.

Training Configurations

- **Config 1**:

Table 5.1: Training configuration 1 for RICE1 experiment

Parameter	**Value**
Dataset size	395 images
Train set size	316 images (80%)
Validation set size	79 images (20%)
Batch size	1
Validation batch size	1
Epochs	200
Learning rate	0.0004

- **Config 2**:

Table 5.2: Training configuration 2 for RICE1 experiment

Parameter	**Value**
Dataset size	395 images
Train set size	316 images (80%)
Validation set size	79 images (20%)
Batch size	1
Validation batch size	1
Epochs	200
Learning rate	0.0004

- **Config 3**:

Table 5.3: Training configuration 3 for RICE1 experiment

Parameter	**Value**
Dataset size	395 images
Train set size	316 images (80%)
Validation set size	79 images (20%)
Batch size	4
Validation batch size	4
Epochs	200
Learning rate	0.0004

- **Config 4**:

Table 5.4: Training configuration 4 for RICE1 experiment

Parameter	**Value**
Dataset size	445 images
Train set size	400 images (90%)
Validation set size	45 images (10%)
Batch size	4
Validation batch size	4
Epochs	200
Learning rate	0.0004

Results

From the RICE1 experiments for training, it was observed that Config 4 resulted in the best overall performance, achieving a PSNR score of 28.87 dB and the highest SSIM of 0.903. This indicated that the reconstructive accuracy and the structural similarity between the images were of high quality. Config 3 also performed well, with a PSNR of 27.40 dB and an SSIM of 0.892. In contrast, Configs 1 and 2 showed lower performance, with PSNR values of 24.98 dB and 23.97 dB, and SSIM values of 0.851 and 0.848, respectively. These results are summarized in the table below.

Table 5.5: PSNR and SSIM results for RICE1 training configurations

Configuration	**PSNR (dB)**	**SSIM**
Config 1	24.98	0.851
Config 2	23.97	0.848
Config 3	27.40	0.892
Config 4	28.87	0.903

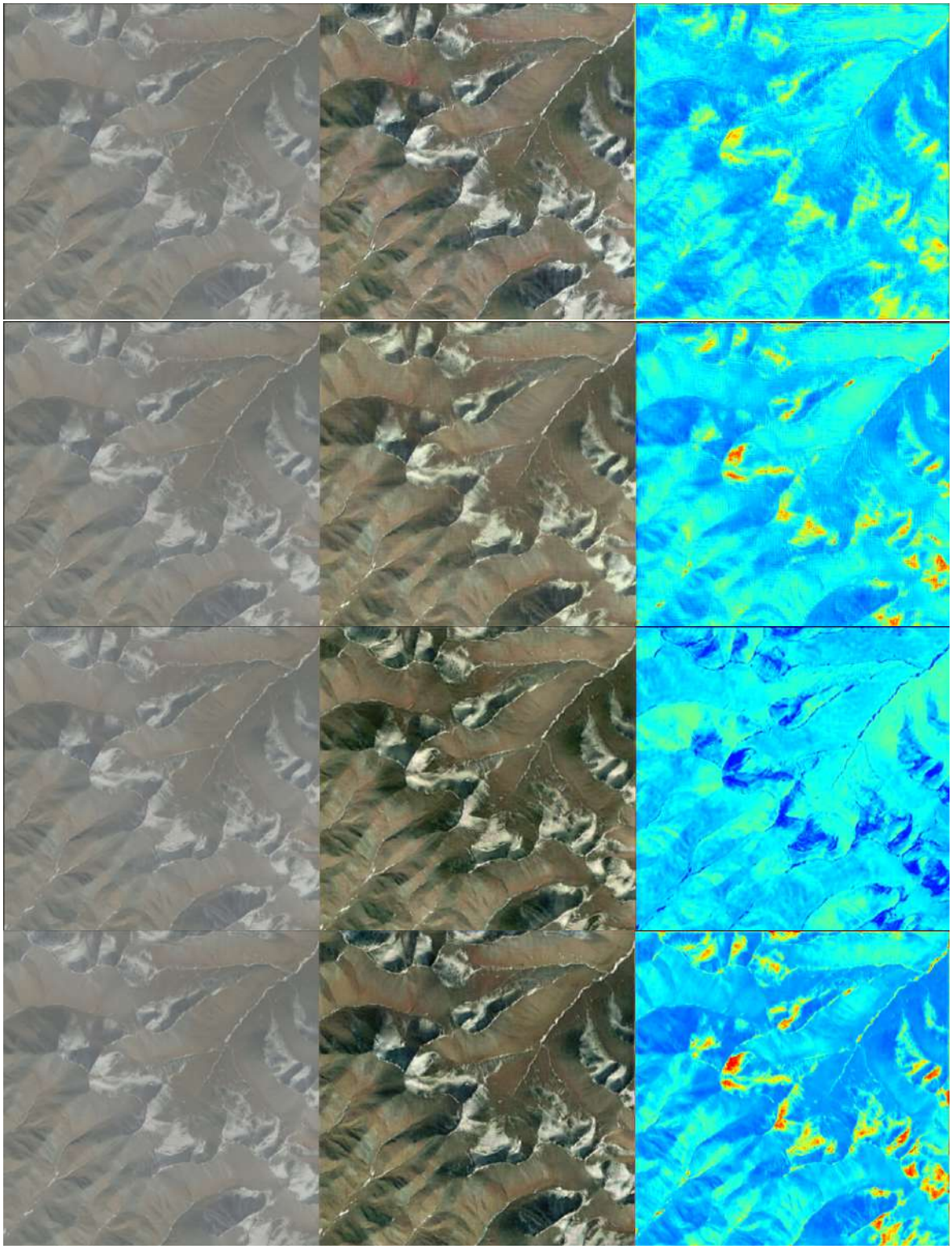


Figure 5.15: Cloud removal results on RICE1 test images using SpA-GAN. Each row corresponds to a different training configuration: Config 1 (top), Config 2 (second), Config 3 (third) and Config 4 (bottom). For each row, the columns show (from left to right): the original cloudy input image, the restored cloud-free output, and the attention heatmap generated by the model.

Performance

The model was able to replicate or improve upon the original SpA-GAN results on RICE1 test images throughout the experiments. The most suitable configuration outperformed the pretrained model in terms of both PSNR and SSIM scores, confirming effective learning of cloud structure and context in the thin cloud domain.

Training with RICE2 (Thick Clouds)

The RICE2 dataset is more complicated, with 646 pairs of images taken from Landsat 8. The main source of the increased difficulty is the presence of thick clouds that completely cover the features of the surface below. This is thus more difficult for the network to solve.

Training Configurations

- **Config 1**:

Table 5.6: Training configuration 1 for RICE2 experiment

Parameter	**Value**
Dataset size	646 images
Train set size	582 images (90%)
Validation set size	64 images (10%)
Batch size	4
Validation batch size	4
Epochs	200
Learning rate	0.0004

- **Config 2**:

Table 5.7: Training configuration 2 for RICE2 experiment

Parameter	**Value**
Dataset size	646 images
Train set size	582 images (90%)
Validation set size	64 images (10%)
Batch size	4
Validation batch size	4
Epochs	300
Learning rate	0.0002

- **Config 3**:

Table 5.8: Training configuration 3 for RICE2 experiment

Parameter	**Value**
Dataset size	646 images
Train set size	582 images (90%)
Validation set size	64 images (10%)
Batch size	4
Validation batch size	4
Epochs	300
Learning rate	0.0004

Results

The results from the RICE2 training experiments show that Config 2 achieved the highest PSNR at 30.54 dB, indicating superior pixel-level accuracy, but had the lowest SSIM at 0.650, reflecting weaker structural preservation. Config 1 gave a more balanced response with a PSNR of 29.63 dB and the highest SSIM of 0.699, which enabled it to have better perceptual quality. Config 3 had lesser results in both parameters, with 29.42 dB PSNR and 0.675 SSIM. The data is presented in the table below.

Table 5.9: PSNR and SSIM results for RICE2 training configurations

Configuration	PSNR (dB)	SSIM
Config 1	29.63	0.699
Config 2	30.54	0.650
Config 3	29.42	0.675

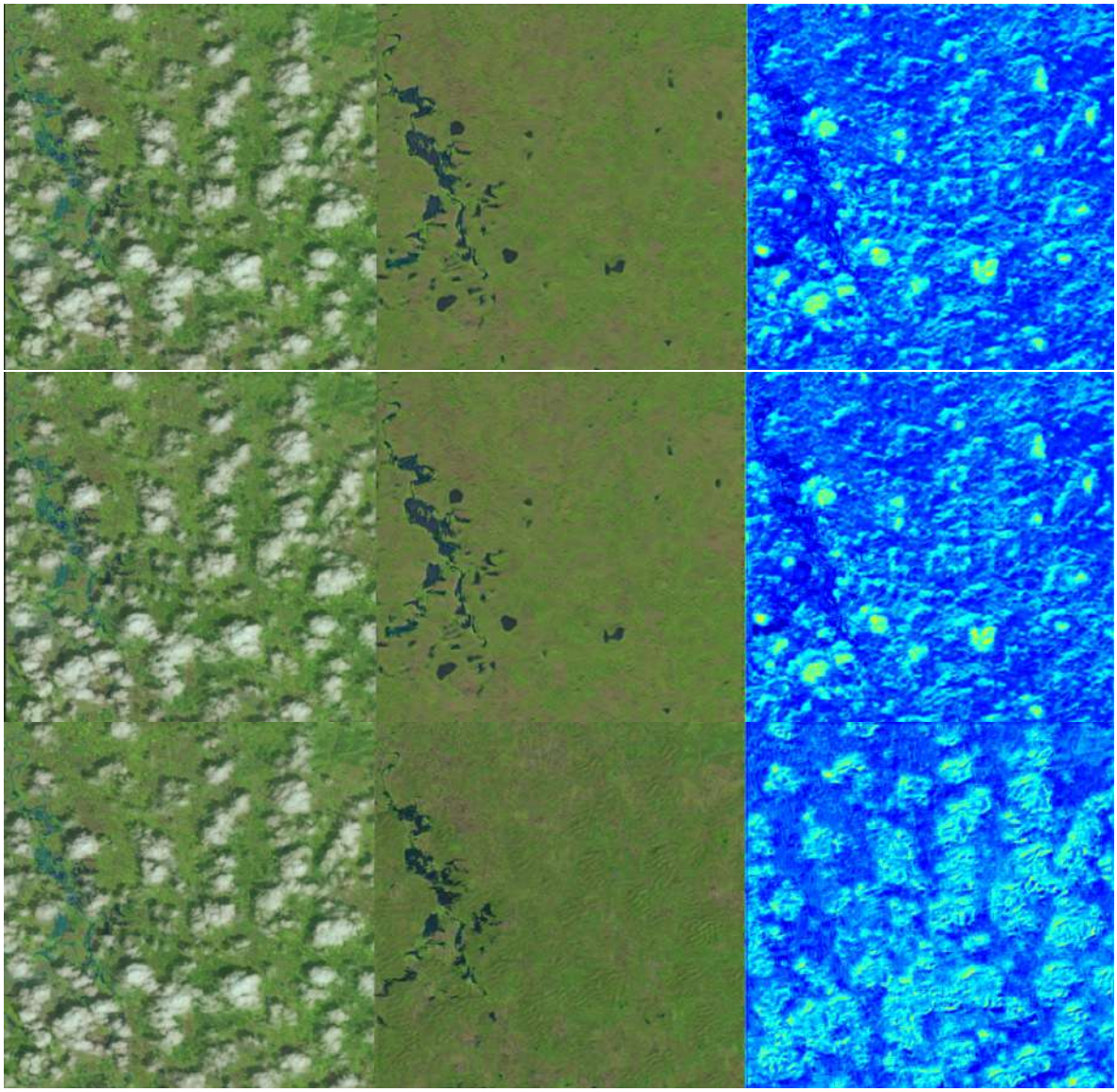


Figure 5.16: Cloud removal results on RICE2 test images using SpA-GAN. Each row corresponds to a different training configuration: Config 1 (top), Config 2 (middle), and Config 3 (bottom). For each row, the columns show (from left to right): the original cloudy input image, the restored cloud-free output, and the attention heatmap generated by the model.

Evaluation and Conclusion

The evaluation results highlight notable differences between the two datasets. For RICE1 that only has sparse cloud coverage, the models achieved higher overall structural quality, and Config 4 was able to get the best SSIM of 0.903 and PSNR of 28.87 dB. This is a sign of efficient reconstruction with strong perceptual fidelity. In contrast, RICE2 presented a more complex challenge due to thicker clouds. Although Config 2 achieved the highest PSNR at 30.54 dB, its SSIM was lower at 0.650, indicating that it was difficult to keep the fine structural details. The maximum SSIM on RICE2 was 0.699 (Config 1), yet still less than RICE1's. These results confirm that while SpA-GAN performs well on thin cloud removal, dense cloud cases call for further improvements to keep the perceptual consistency.

5.3 SpA-GAN Model on Synthetic Flare Dataset

This section describes the main training phase of the model using the synthetic flare dataset. The goal of this phase was to fine-tune the GAN for the task of lens flare removal, utilizing the synthetic dataset generated in earlier stages. The training dataset consists of paired images, where each image is affected by lens flare, along with its corresponding ground truth image, which is free from such artifacts. These paired images provide the model with a clear target for learning the mapping from flare-degraded images to their restored versions. Various configurations and parameters were explored during the training process, including adjustments to the learning rate, batch size, and number of epochs.

In some of the training phases, the best weights for the generator and discriminator from previous training runs were used to initialize the models. This allowed the system to build upon the progress made in earlier stages, ensuring a smoother transition and faster convergence. Additionally, a Learning Rate Scheduler was employed to adjust the learning rate dynamically during training, which helped optimize the model's performance. The VGG Perceptual Loss was also used in the training process to improve the perceptual quality of the restored images, by guiding the network to generate images that are more visually similar to the ground truth.

Learning Rate Scheduler

In the training process, the Learning Rate Scheduler was utilized to dynamically adjust the learning rate for both the generator and discriminator optimizers. This approach ensures that the model converges efficiently while avoiding issues like overshooting the optimal solution. The scheduler helps maintain a stable learning rate throughout the training process by reducing it at specific intervals.

For both the generator and discriminator, the StepLR scheduler was employed. This scheduler reduces the learning rate by a factor of `gamma` every `step_size` epochs. Specifically, the `step_size` and `gamma` values varied between different training configurations to optimize the training process. For instance, in some configurations, the learning rate was reduced every 20 epochs by a factor of 0.5, while in other configurations, the reduction occurred with a different step size and gamma value. This adjustment allowed the model to fine-tune its parameters gradually and effectively during training, ultimately helping to achieve better performance.

The use of a dynamic learning rate scheduler played a crucial role in improving model convergence and performance, especially in later training stages where smaller learning rates are needed for more precise adjustments to the model weights.

VGG Perceptual Loss

The VGG Perceptual Loss was incorporated into the training process to improve the perceptual quality of the restored images. Perceptual loss is based on high-level feature comparisons rather than pixel-wise differences, allowing the model to focus on the structural and semantic content of the image. This approach makes the restoration more visually realistic and aligned with human perception.

The VGG Perceptual Loss uses a pre-trained VGG16 model, which is commonly used in image restoration tasks due to its ability to capture relevant high-level features. Specifically, the layers up to `relu2_2` of the VGG16 architecture are employed, which corresponds to the 9th layer of the model. These layers are essential for capturing texture and structural details, which are important for the perceptual quality of restored images.

The VGG model is initialized with pre-trained weights from ImageNet, ensuring that it has already learned robust feature representations. During training, the VGG model's parameters are frozen, meaning they are not updated during backpropagation. This allows the model

to retain the pre-learned features from the ImageNet dataset, which are then applied to the flare restoration task.

Additionally, during the training phases where VGG Perceptual Loss was used, the weight of the perceptual loss, denoted as λ_{percep}, was varied across different training configurations. This variation allowed for fine-tuning the influence of the perceptual loss on the overall training process. In some configurations, a smaller λ_{percep} (e.g., 0.01) was used, while in others, a larger value (e.g., 0.05) was applied. By adjusting λ_{percep}, the model was able to balance between achieving pixel-wise accuracy and maintaining perceptual quality, leading to improved restoration results.

The training configurations used during the process are listed below, providing a detailed overview of each setup.

Training 1 (0028)

Table 5.10: Training Configuration 1

Parameter	**Value**
Dataset	`Dataset1` (train, validation with 80-20 split)
Learning Rate	0.0004
Epochs	300
Snapshot Interval	Every 50 epochs
Pretrained Weights Gen	-
Pretrained Weights Dis	-
Learning Rate Scheduler	-
VGG Perceptual Loss	-

Training 2 (0029)

Table 5.11: Training Configuration 2

Parameter	**Value**
Dataset	`Dataset2` (train, validation with 80-20 split)
Learning Rate	0.0004
Epochs	300
Snapshot Interval	Every 50 epochs
Pretrained Weights Gen	-
Pretrained Weights Dis	-
Learning Rate Scheduler	StepLR, step_size=100, gamma=0.8
VGG Perceptual Loss	-

Training 3 (0030)

Table 5.12: Training Configuration 3

Parameter	**Value**
Dataset	`Dataset3` (train, validation with 80-20 split)
Learning Rate	0.0001
Epochs	150
Snapshot Interval	Every 25 epochs
Pretrained Weights Gen	`best_generator.pth` from training 2
Pretrained Weights Dis	`my_train_dis_model_epoch_300.pth` from training 2
Learning Rate Scheduler	StepLR, step_size=100, gamma=0.8
VGG Perceptual Loss	-

Training 4 (0035)

Table 5.13: Training Configuration 4

Parameter	**Value**
Dataset	`Dataset2.5` (train, validation with 80-20 split)
Learning Rate	0.0001
Epochs	100
Snapshot Interval	Every 25 epochs
Pretrained Weights Gen	`best_generator.pth` from training 3
Pretrained Weights Dis	`my_train_dis_model_epoch_150.pth` from training 3
Learning Rate Scheduler	StepLR, step_size=100, gamma=0.8
VGG Perceptual Loss	-

Training 5 (0037)

Table 5.14: Training Configuration 5

Parameter	**Value**
Dataset	`Dataset3` (train, validation with 80-20 split)
Learning Rate	0.00005
Epochs	50
Snapshot Interval	Every 25 epochs
Pretrained Weights Gen	`best_generator.pth` from training 4
Pretrained Weights Dis	`my_train_dis_model_epoch_150.pth` from training 4
Learning Rate Scheduler	StepLR, step_size=100, gamma=0.8
VGG Perceptual Loss	$\lambda_{\text{percep}} = 0.01$

Training 6 (0039)

Table 5.15: Training Configuration 6

Parameter	**Value**
Dataset	`Dataset3` (train, validation with 80-20 split)
Learning Rate	0.00005
Epochs	50
Snapshot Interval	Every 25 epochs
Pretrained Weights Gen	`best_generator.pth` from training 5
Pretrained Weights Dis	`my_train_dis_model_epoch_50.pth` from training 5
Learning Rate Scheduler	StepLR, step_size=100, gamma=0.8
VGG Perceptual Loss	$\lambda_{\text{percep}} = 0.01$

Training 7 (0041)

Table 5.16: Training Configuration 7

Parameter	**Value**
Dataset	`Dataset3` (train, validation with 80-20 split)
Learning Rate	0.0001 (with scheduler)
Epochs	100
Snapshot Interval	Every 25 epochs
Pretrained Weights Gen	`best_generator.pth` from training 5
Pretrained Weights Dis	`my_train_dis_model_epoch_50.pth` from training 5
Learning Rate Scheduler	StepLR, step_size=20, gamma=0.5
VGG Perceptual Loss	$\lambda_{\text{percep}} = 0.05$

5.4 Real-Time Demo Implementation

The real-time demonstration of flare restoration was implemented using a combination of OpenCV, PyTorch, and the SpA-GAN model. The system continuously monitors a specified

directory for new images to process. As new images are added to the directory, they are automatically loaded, preprocessed, and passed through the trained SpA-GAN model for flare restoration. The process is designed to run in real-time, enabling immediate visualization of the restored images.

It is important to note that some delay between frame updates may occur during the demo. This is primarily due to the fact that both the GAN inference and the Unreal Engine rendering processes are executed on the same system. As both tasks are computationally intensive—especially when using high-resolution scenes and deep neural networks—the system load can cause brief latency in image acquisition and restoration.

5.4.1 System Workflow

The system operates as follows:

- **Environment Simulation and Image Capture**: The process begins with the Unreal Engine environment, where an AI-controlled player continuously follows a predefined path. As it moves through the virtual scene, it captures images in real time and saves them to a designated output folder.
- **Directory Monitoring**: This designated folder is constantly monitored for new images, specifically PNG files. As soon as a new image is detected, it is added to the processing queue.
- **Image Loading and Preprocessing**: Each new image is loaded using OpenCV's `cv2.imread()` function. The image is resized to match the required dimensions for the model, normalized, and converted into a tensor compatible with PyTorch.
- **Model Inference**: The preprocessed image is passed through the SpA-GAN generator model, which was pretrained and loaded into the system. The model performs a forward pass using PyTorch, generating a restored (flare-free) image.
- **Real-Time Display**: The restored image is displayed in real-time using OpenCV's `cv2.imshow()` function. This allows the user to visually inspect the flare removal results as they occur.

A video demonstration of the real-time flare removal system can be viewed at the following link: **Real-Time Flare Removal Demo using GAN**.

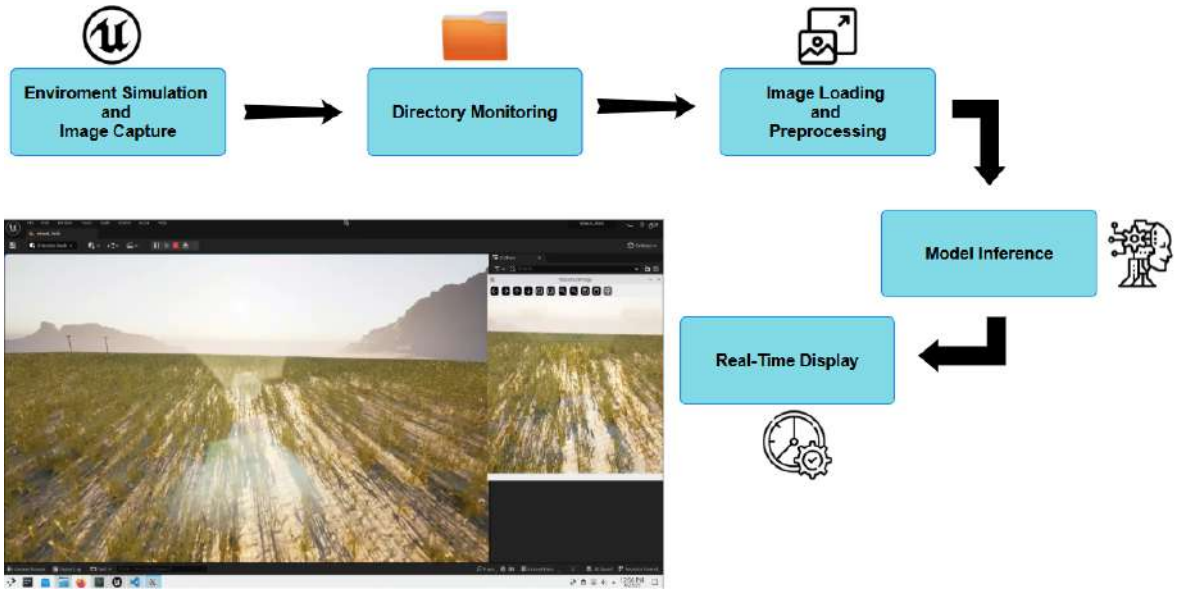


Figure 5.17: Real-Time Flow Chart.

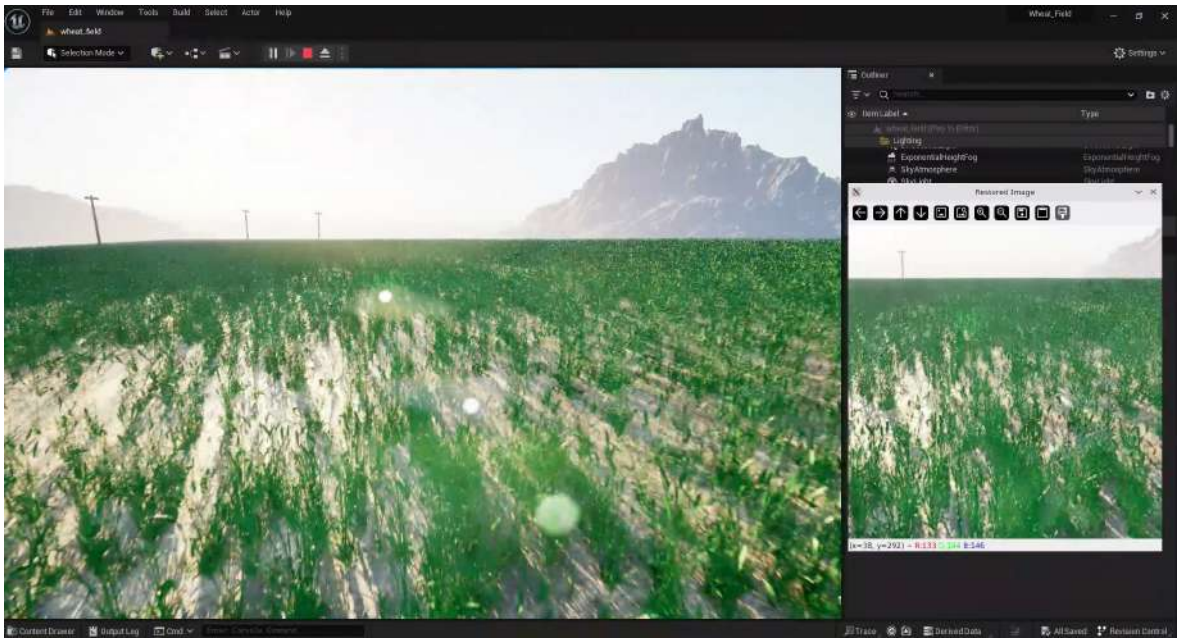


Figure 5.18: Real-Time Demo Implementation.

Chapter 6

Evaluation

6.1 Quantitative Evaluation

This section presents the quantitative assessment of the model's performance using various metrics. We evaluated the model's ability to restore vegetation features and the quality of the images through several performance metrics as outlined in the Background chapter.

6.1.1 Metrics Used for Evaluation

For this evaluation, the following metrics were used to quantify the model's performance:

- **PSNR (Peak Signal-to-Noise Ratio)**: A measure of image quality that compares the difference between the restored image and the ground truth.
- **SSIM (Structural Similarity Index)**: A metric for perceptual similarity considering luminance, contrast, and structure.
- **GRVI (Green-Red Vegetation Index)**: Used to assess the preservation of vegetation features by comparing the green and red channel intensities.

These metrics are applied to evaluate the quality of the restored images and to verify that vegetation structures are properly preserved during the restoration process. The results of these evaluations are presented and analyzed in the following sections.

6.1.2 PSNR Evaluation

PSNR provides a quantitative measure of image restoration quality. It compares the pixel-by-pixel difference between the restored image and the ground truth, indicating how well the model has preserved the details of the image.

Training Results

Table 6.1: PSNR Results for all Training Configurations

Configuration	PSNR (dB)
Training 1 (0028)	24.5972
Training 2 (0029)	27.9549
Training 3 (0030)	28.3847
Training 4 (0035)	28.4446
Training 5 (0037)	28.7941
Training 6 (0039)	28.5202
Training 7 (0041)	28.6480

The PSNR values show a clear upward trend as the model progresses through each training phase. Initially, in **Training 1**, the PSNR value is **24.5972**, indicating a relatively low image restoration quality with noticeable distortions. A notable leap occurs in **Training 2**, where the PSNR jumps to **27.9549**. This sharp increase of over 3 dB suggests that the model has rapidly internalized fundamental patterns for distinguishing flare-affected regions and has begun producing more visually and structurally consistent reconstructions. Such a gain typically corresponds to a significant qualitative improvement in perceived image quality.

In subsequent training phases, from **Training 3** to **Training 5**, the PSNR values continue to rise steadily, reaching a maximum of **28.7941** in **Training 5**. These modest but consistent improvements indicate that the model is transitioning from basic reconstruction toward fine-tuning—learning to reduce residual errors and better preserve subtle image structures.

Beyond **Training 5**, the PSNR values plateau, with **Training 6** and **Training 7** achieving **28.5202** and **28.6480**, respectively. These slight fluctuations reflect the model's stabilization, indicating that it has reached a convergence point where further learning yields diminishing returns in terms of pixel-level improvement.

The extremely constrained range of PSNR results following Training 3, all greater than 28 dB, reflects the fact that the model consistently produces high-quality outputs after that point.

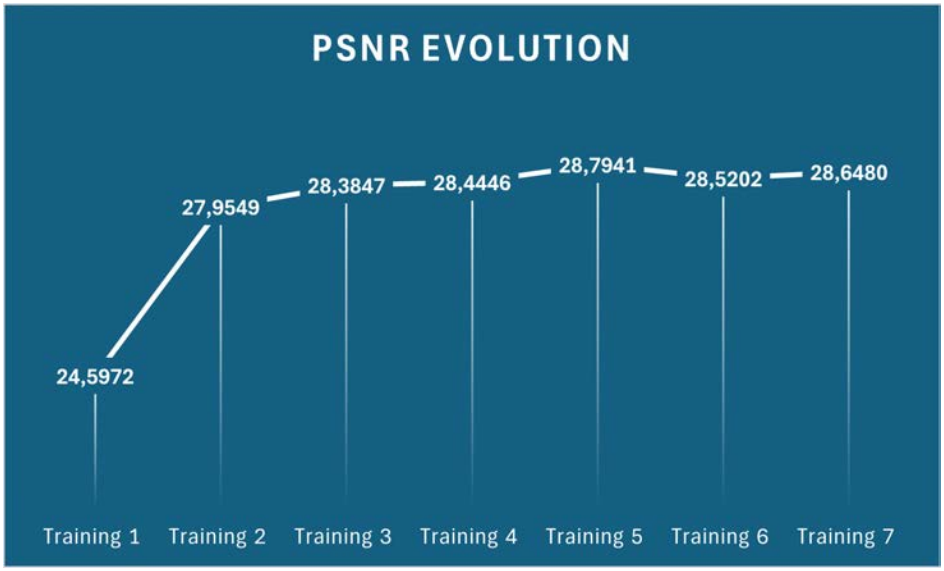


Figure 6.1: PSNR Evolution during trainings.

In practice, PSNR values in the 28–29 dB range are considered very good for image restoration tasks, especially when dealing with challenging artifacts like lens flare. This validates the effectiveness of the proposed approach and also emphasizes the importance of extended training to achieve reliable and stable performance.

6.1.3 SSIM Evaluation

SSIM is a metric that measures the similarity between two images, focusing on structural information, luminance, and texture. Unlike pixel-based metrics, SSIM evaluates changes that align with human perception, making it a better indicator of visual quality. Higher SSIM values indicate a closer match between the restored image and the ground truth, reflecting better preservation of important features like edges and textures.

Training Results

Table 6.2: SSIM Results for all Training Configurations

Configuration	SSIM
Training 1 (0028)	0.9105
Training 2 (0029)	0.9185
Training 3 (0030)	0.9202
Training 4 (0035)	0.9201
Training 5 (0037)	0.9215
Training 6 (0039)	0.9206
Training 7 (0041)	0.9198

The SSIM values exhibit a clear upward trend as the model progresses through each training phase. Initially, in **Training 1**, the SSIM value is **0.9105**, reflecting a relatively high structural similarity between the original and restored images. By **Training 2**, the SSIM improves to **0.9185**, indicating a noticeable enhancement in the model's ability to preserve structural features in the restored images.

In subsequent training phases, from **Training 3** to **Training 5**, the SSIM values continue to rise steadily, reaching a maximum of **0.9215** in **Training 5**. This gradual increase, though numerically modest, signifies a meaningful improvement in the model's ability to reconstruct fine-grained structural details—an aspect critical to the perceived visual quality of the restored outputs. The peak at **Training 5** corresponds to the point at which the model achieves its best performance in terms of structural consistency with the original image.

After this peak, there is minimal variation in the SSIM values, with **Training 6** and **Training 7** scoring **0.9206** and **0.9198**, respectively. The small fluctuations show that a plateau has been reached in the learning of the model, after which further training produces minimal increases in structural accuracy. Although the values plateau, they remain high throughout the final phases, again reinforcing the consistency of the model in maintaining image geometry, edges, and textural integrity.

Figure 6.2: SSIM Evolution during trainings.

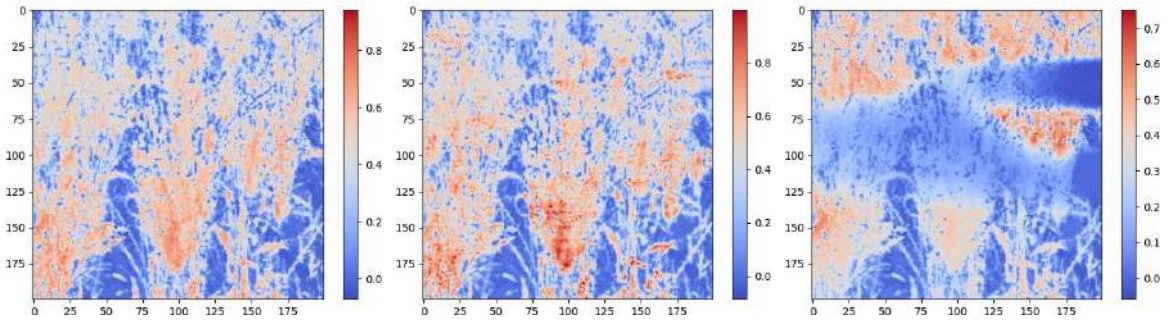
In summary, the SSIM evolution across trainings highlights the model's increasing ability to preserve structural features, which is vital for realistic image restoration. While the improvements become less pronounced in the later stages, this consistent upward trend highlights that the model is continuously learning to improve the structural similarity between the original and restored images.

6.1.4 GRVI Evaluation

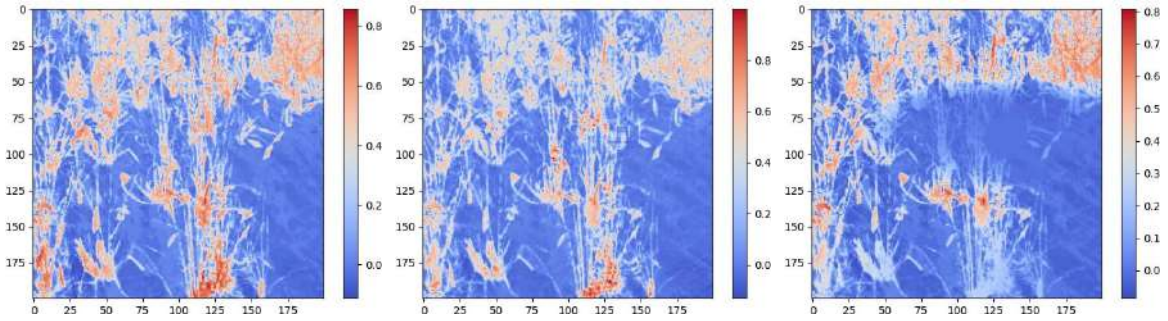
This subsection evaluates the **Green-Red Vegetation Index (GRVI)**, which is used to assess the vegetation content in the restored images. GRVI is calculated for the ground truth, restored, and flared images. We compute the average GRVI over a selected region of interest (ROI) in each image. The ROI is a rectangular region of size 200 × 200 pixels, positioned in the lower half of each image, centered horizontally and starting from the vertical center of the image extending downward. This region targets the ground-level area where vegetation is most concentrated and where lens flare artifacts typically have the most significant impact on vegetation index calculations. These GRVI values are compared to assess how well the model has preserved the vegetation features.

GRVI is particularly well-suited for analyzing green grass vegetation because it specifically exploits the spectral characteristics of healthy grass. Green grass exhibits high reflectance in the green spectral band due to chlorophyll's selective absorption properties, while showing relatively lower reflectance in the red band where chlorophyll absorption is stronger.

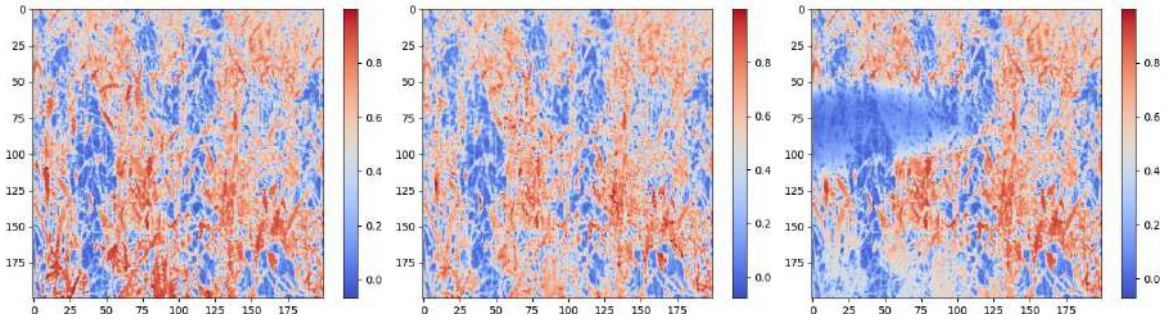
The GRVI formula effectively captures this spectral contrast, making it more sensitive to the presence and health of grass vegetation compared to other vegetation indices that might be optimized for different types of vegetation or broader spectral ranges.



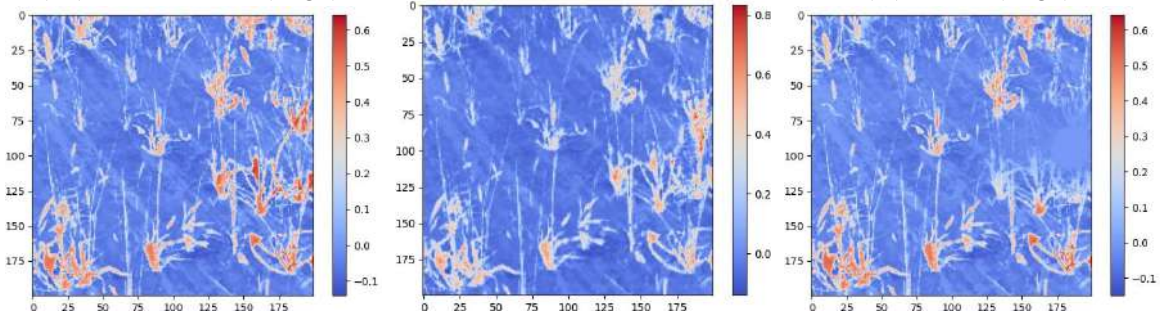
(i) Ground Truth (img1). (ii) Restored (img1). (iii) Flared (img1).



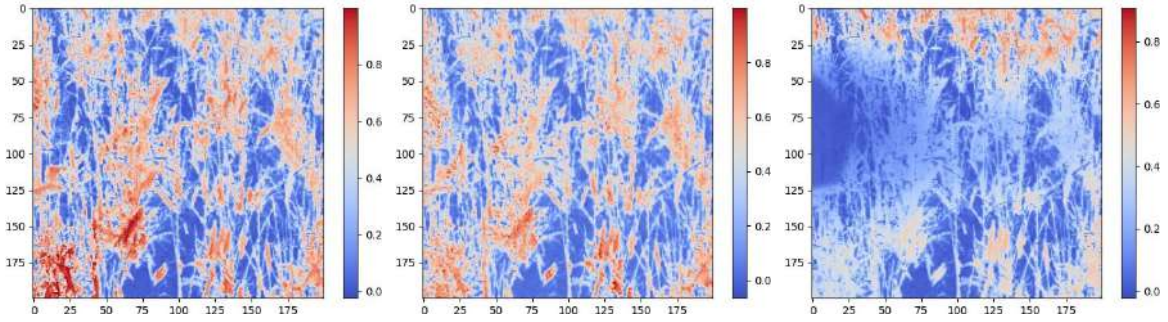
(iv) Ground Truth (img2). (v) Restored (img2). (vi) Flared (img2).



(vii) Ground Truth (img3). (viii) Restored (img3). (ix) Flared (img3).

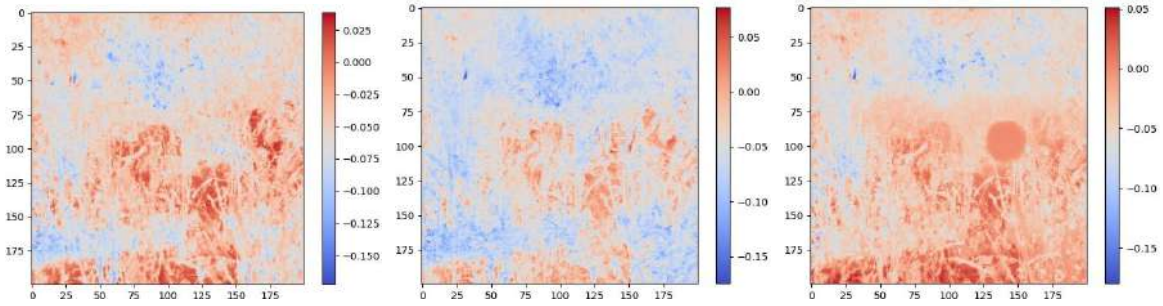


(x) Ground Truth (img4). (xi) Restored (img4). (xii) Flared (img4).

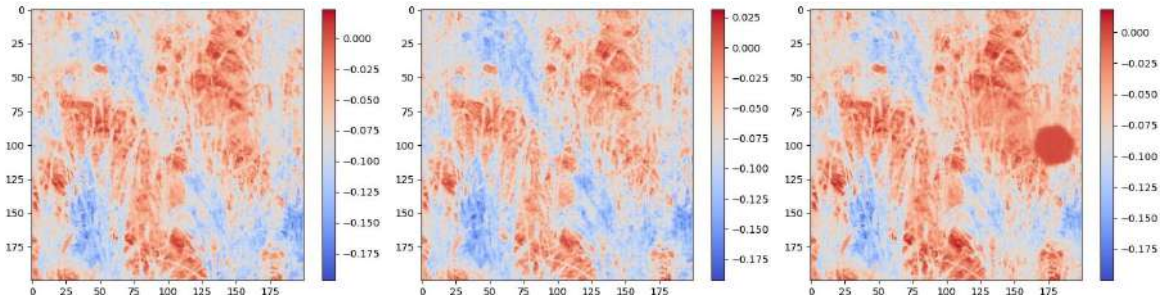


(xiii) Ground Truth (img5). (xiv) Restored (img5). (xv) Flared (img5).

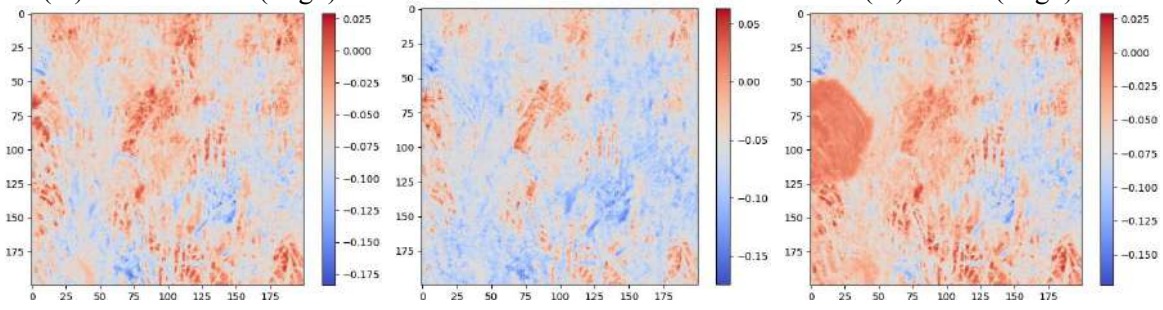
Figure 6.3: GRVI results in green grass images.



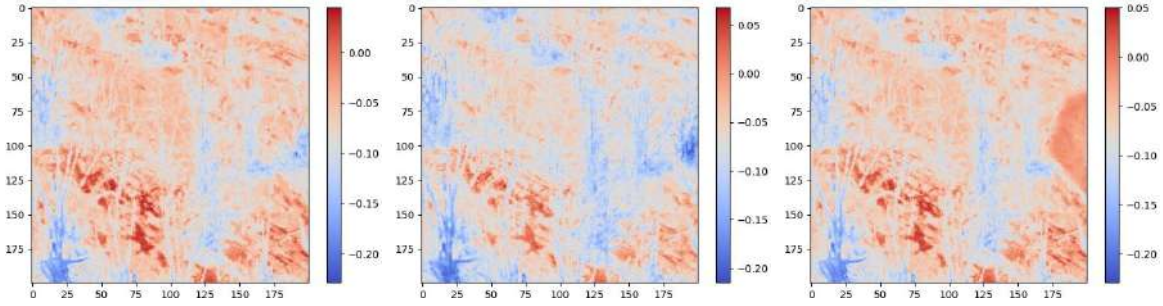
(i) Ground Truth (img6). (ii) Restored (img6). (iii) Flared (img6).



(iv) Ground Truth (img7). (v) Restored (img7). (vi) Flared (img7).



(vii) Ground Truth (img8). (viii) Restored (img8). (ix) Flared (img8).



(x) Ground Truth (img9). (xi) Restored (img9). (xii) Flared (img9).

Figure 6.4: GRVI results in dry grass images.

As seen in the images, the GRVI evaluation confirms that the restoration method effectively preserves vegetation spectral characteristics while removing lens flare artifacts. The close correspondence between GRVI values in restored and ground truth images demonstrates that the method maintains the spectral integrity essential for accurate grass vegetation analysis, making it suitable for applications requiring both visual enhancement and quantitative vegetation assessment. To further support these observations, two GRVI charts were generated, comparing the average GRVI values across three categories: ground truth (flare-free), restored and flared images.

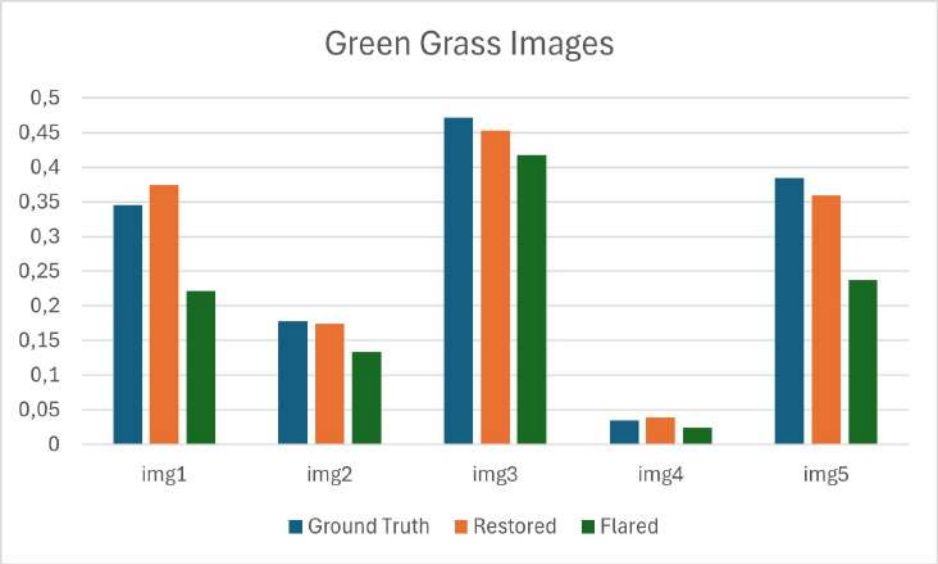


Figure 6.5: Comparison of GRVI values across ground truth, restored and flared green grass images.

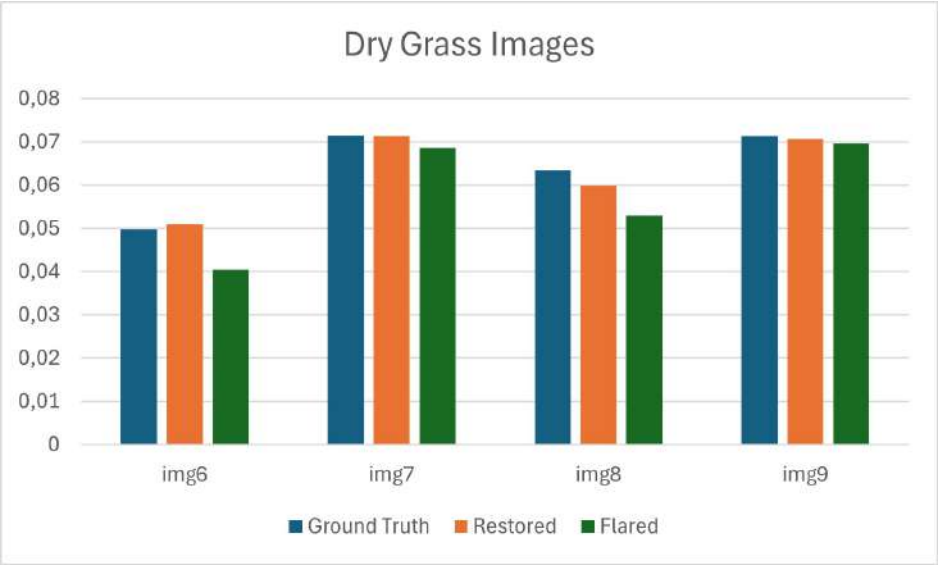


Figure 6.6: Comparison of GRVI values across ground truth, restored and flared dry grass images.

Ground truth images exhibit the highest GRVI values, as expected, reflecting strong vegetation signals. In contrast, flared images show reduced GRVI values due to the distortion and loss of spectral information caused by lens flare. The most important thing is that the restored images recover much of the lost information, with GRVI values significantly improved compared to the flare inputs and closely aligned with the ground truth. These results quantitatively validate the visual improvement and confirm that the proposed method restores not only appearance but also the underlying spectral information critical for vegetation monitoring.

6.2 Qualitative Evaluation

In this section, we present a qualitative evaluation of the model's performance across all training configurations by visually comparing the flared images, restored images, and the ground truth. The goal is to assess how effectively the model restores images affected by flare, focusing on the preservation of image quality and key features, especially vegetation structures. For each training, the model's output is compared with the original, unflared ground truth to identify areas where the flare removal process succeeded or caused distortions. The flared images show the degradation caused by lens flare, while the restored images demonstrate the model's ability to remove or reduce the flare, bringing the image closer to the ground

truth. The results are presented for each training configuration, highlighting the effectiveness of the model in different scenarios and showing how the training process improves the flare removal quality.

Figure 6.7: Qualitative comparison of flared, restored, and ground truth images for Training 1 to 7. Each row shows the flared image, restored image, and ground truth, respectively.

Figure 6.8: Qualitative comparison of flared, restored, and ground truth images for Training 1 to 7. Each row shows the flared image, restored image, and ground truth, respectively.

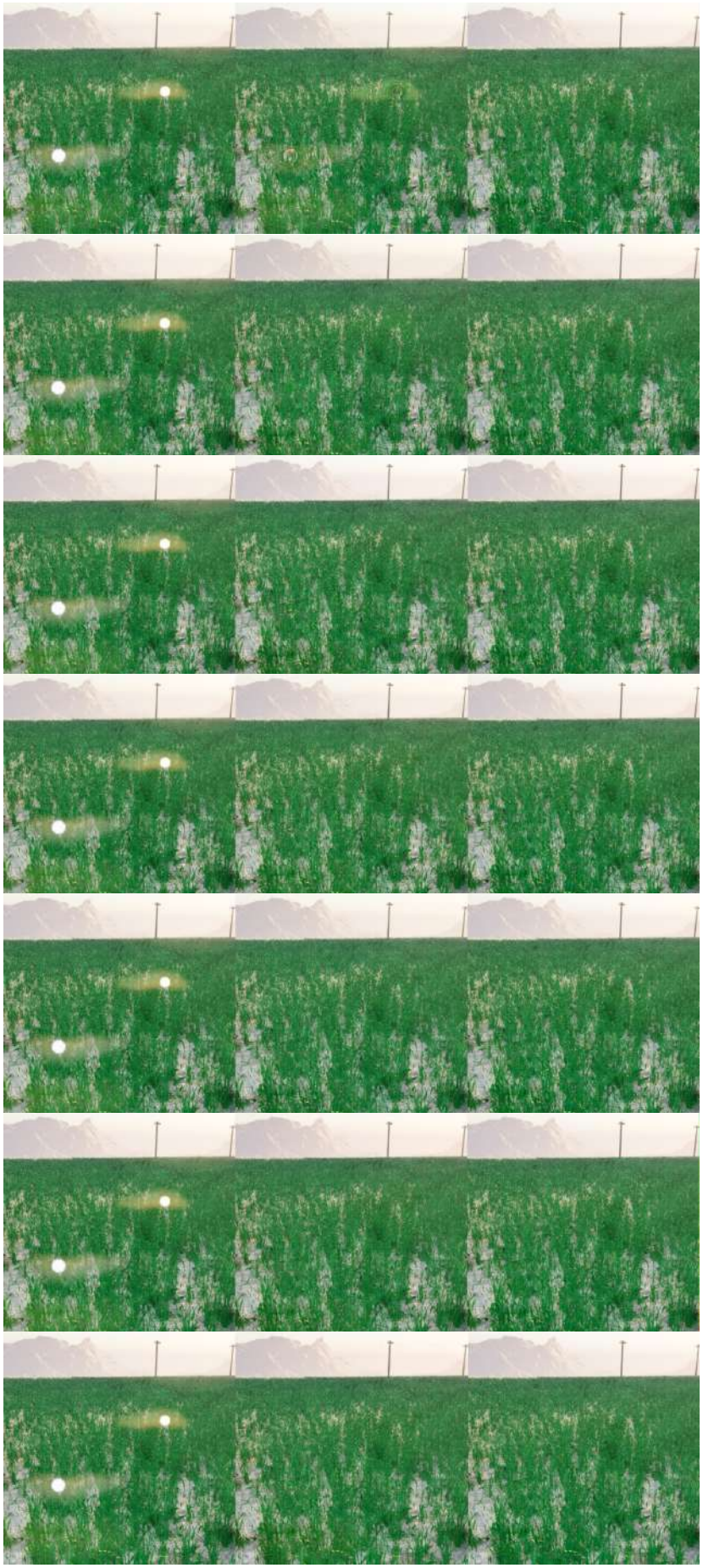


Figure 6.9: Qualitative comparison of flared, restored, and ground truth images for Training 1 to 7. Each row shows the flared image, restored image, and ground truth, respectively.

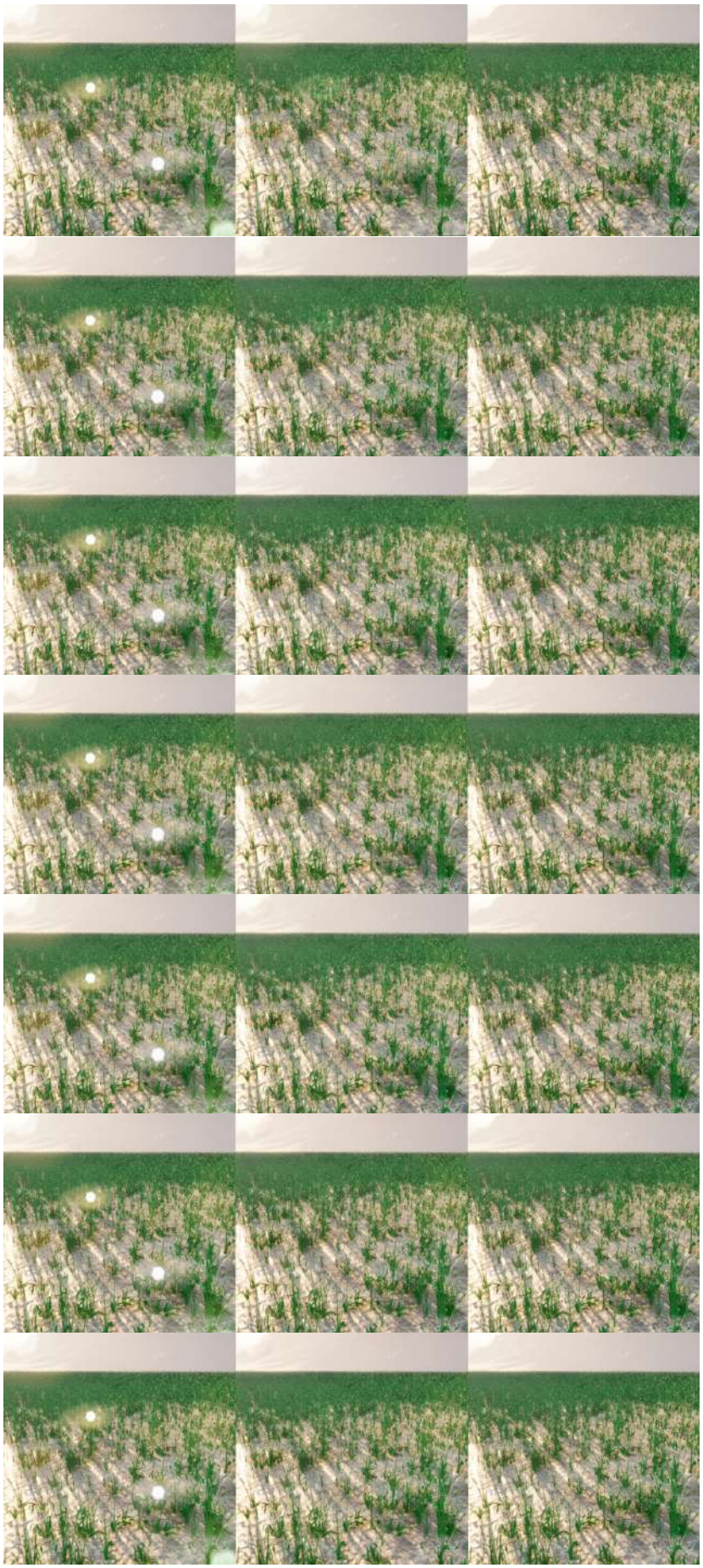


Figure 6.10: Qualitative comparison of flared, restored, and ground truth images for Training 1 to 7. Each row shows the flared image, restored image, and ground truth, respectively.

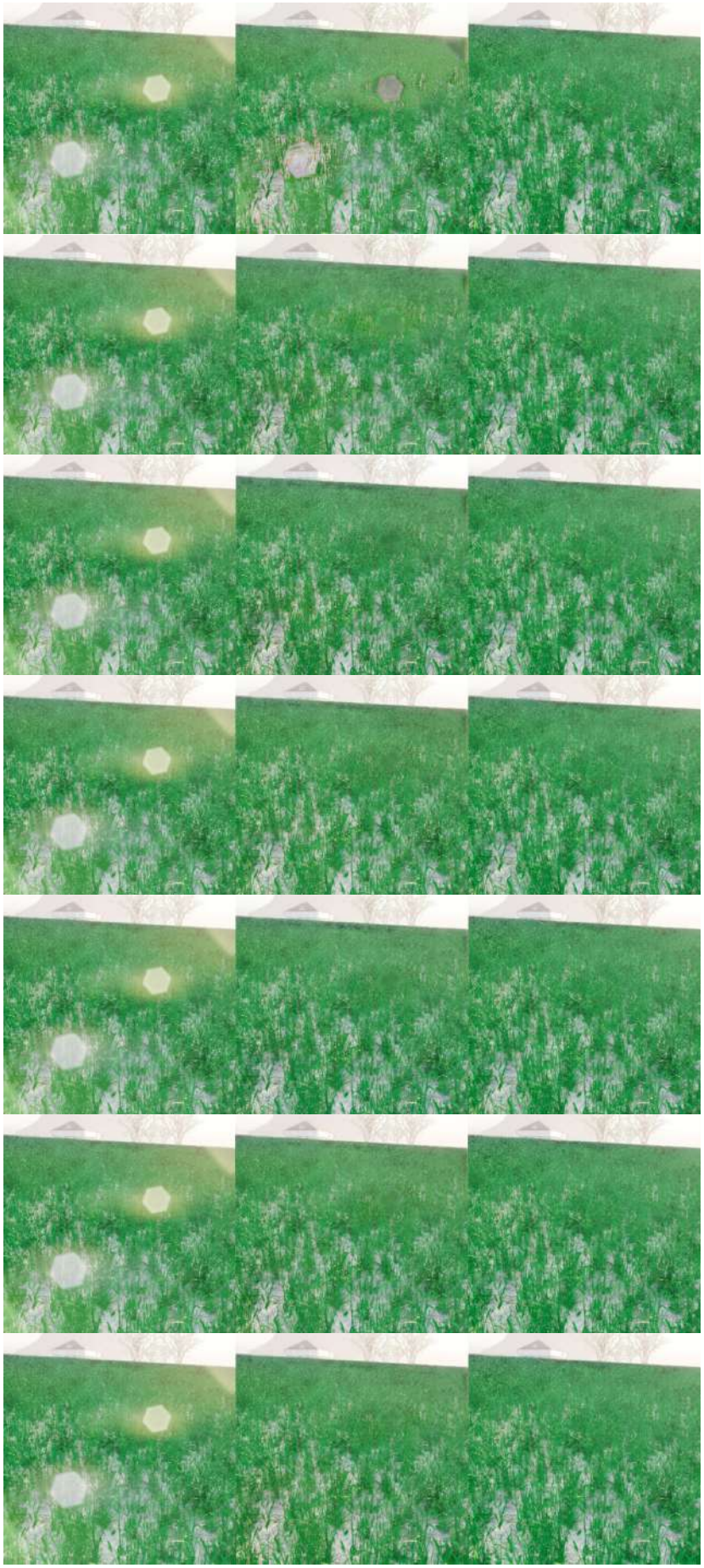


Figure 6.11: Qualitative comparison of flared, restored, and ground truth images for Training 1 to 7. Each row shows the flared image, restored image, and ground truth, respectively.

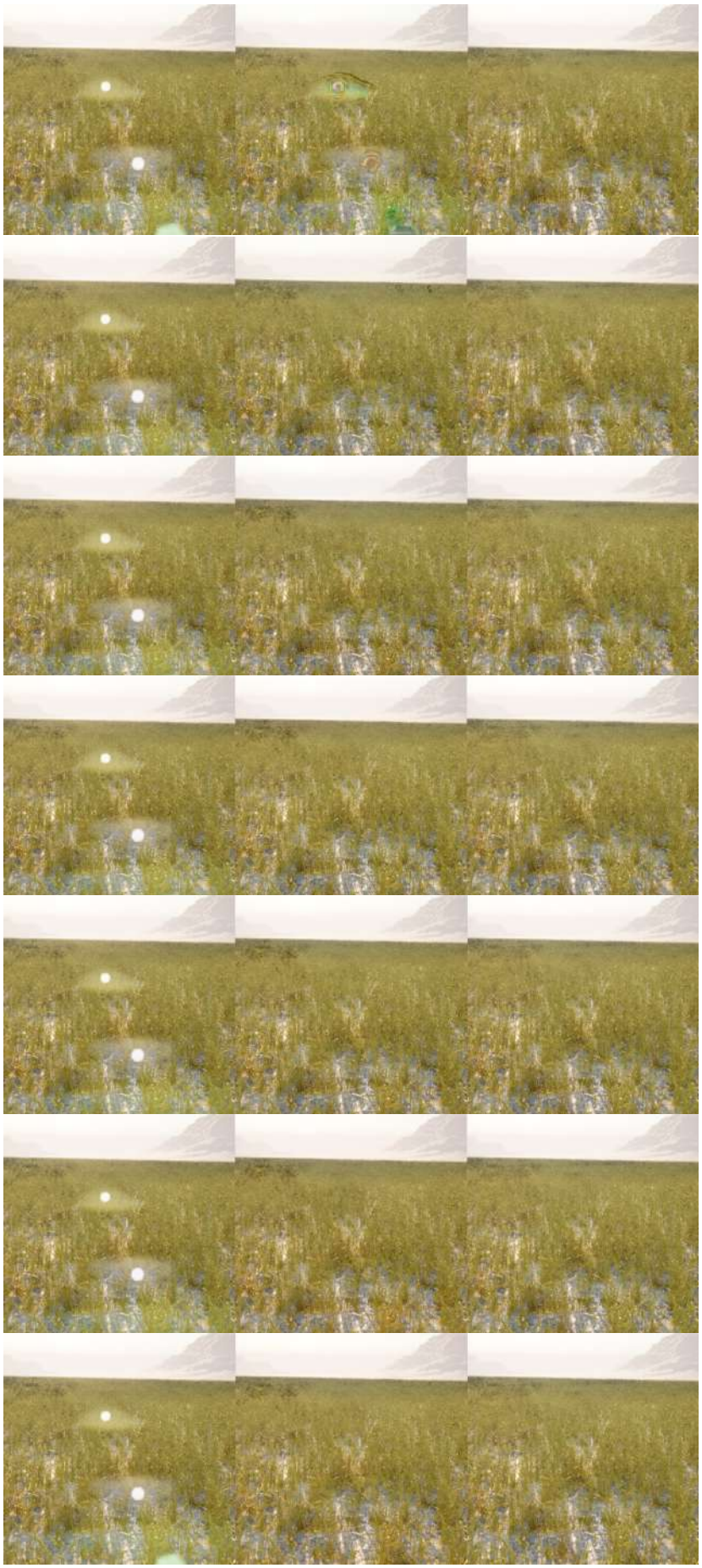


Figure 6.12: Qualitative comparison of flared, restored, and ground truth images for Training 1 to 7. Each row shows the flared image, restored image, and ground truth, respectively.

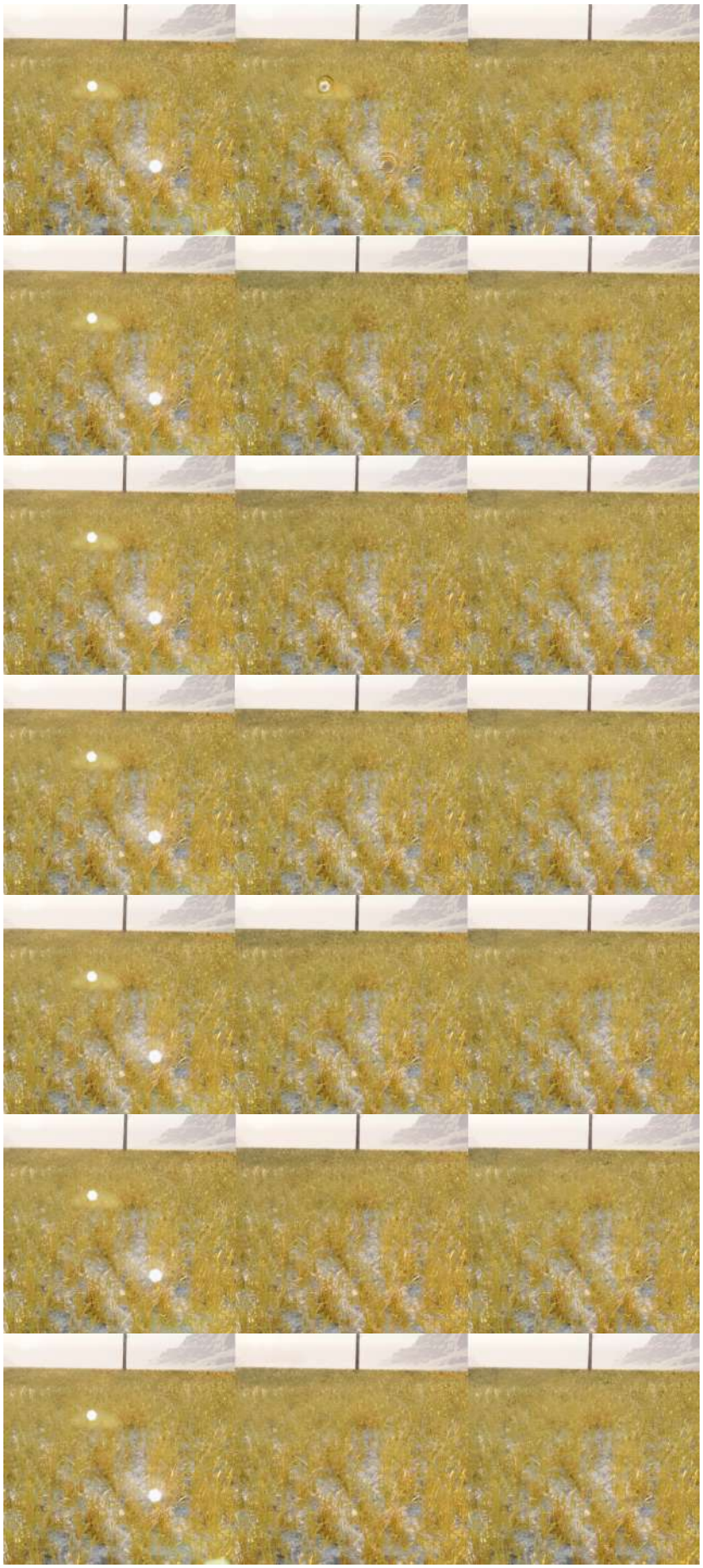


Figure 6.13: Qualitative comparison of flared, restored, and ground truth images for Training 1 to 7. Each row shows the flared image, restored image, and ground truth, respectively.

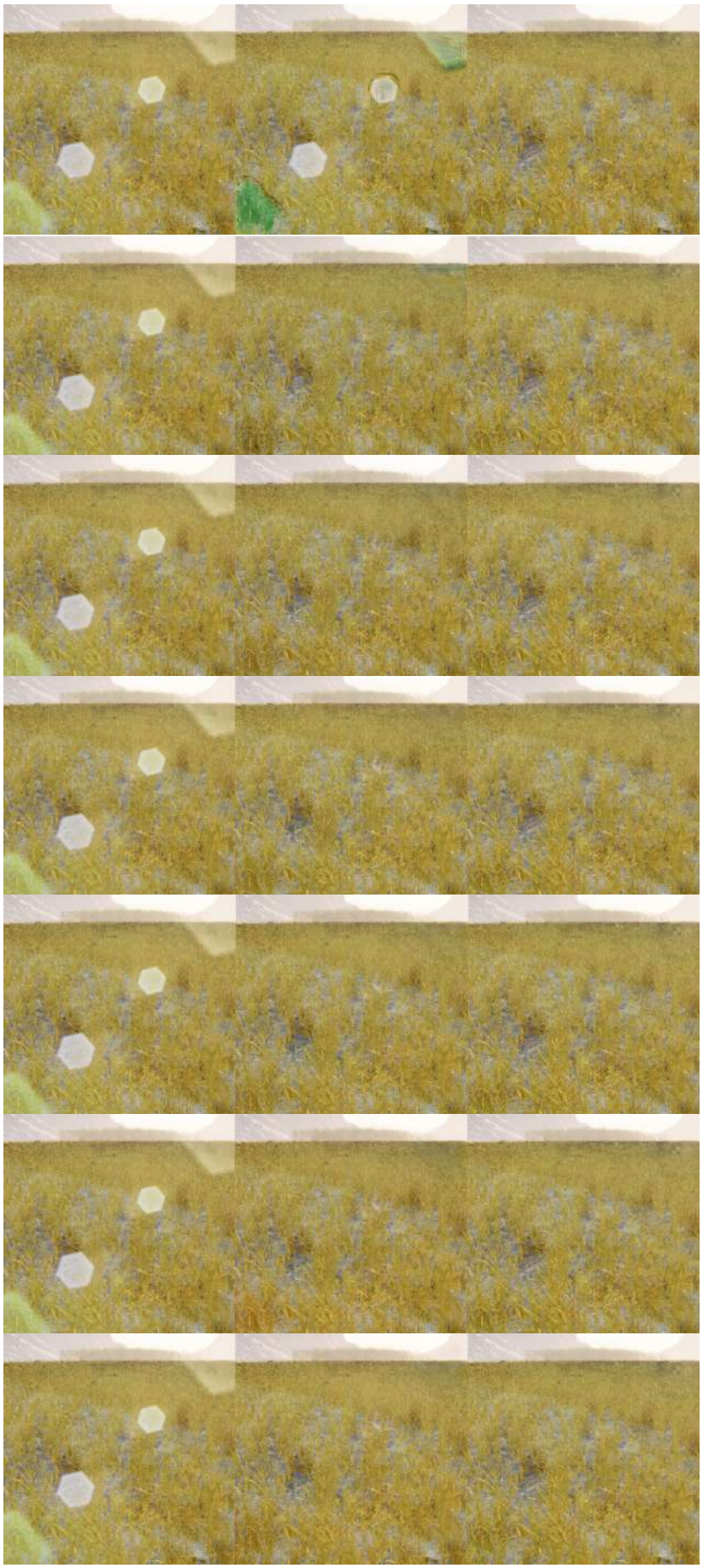


Figure 6.14: Qualitative comparison of flared, restored, and ground truth images for Training 1 to 7. Each row shows the flared image, restored image, and ground truth, respectively.

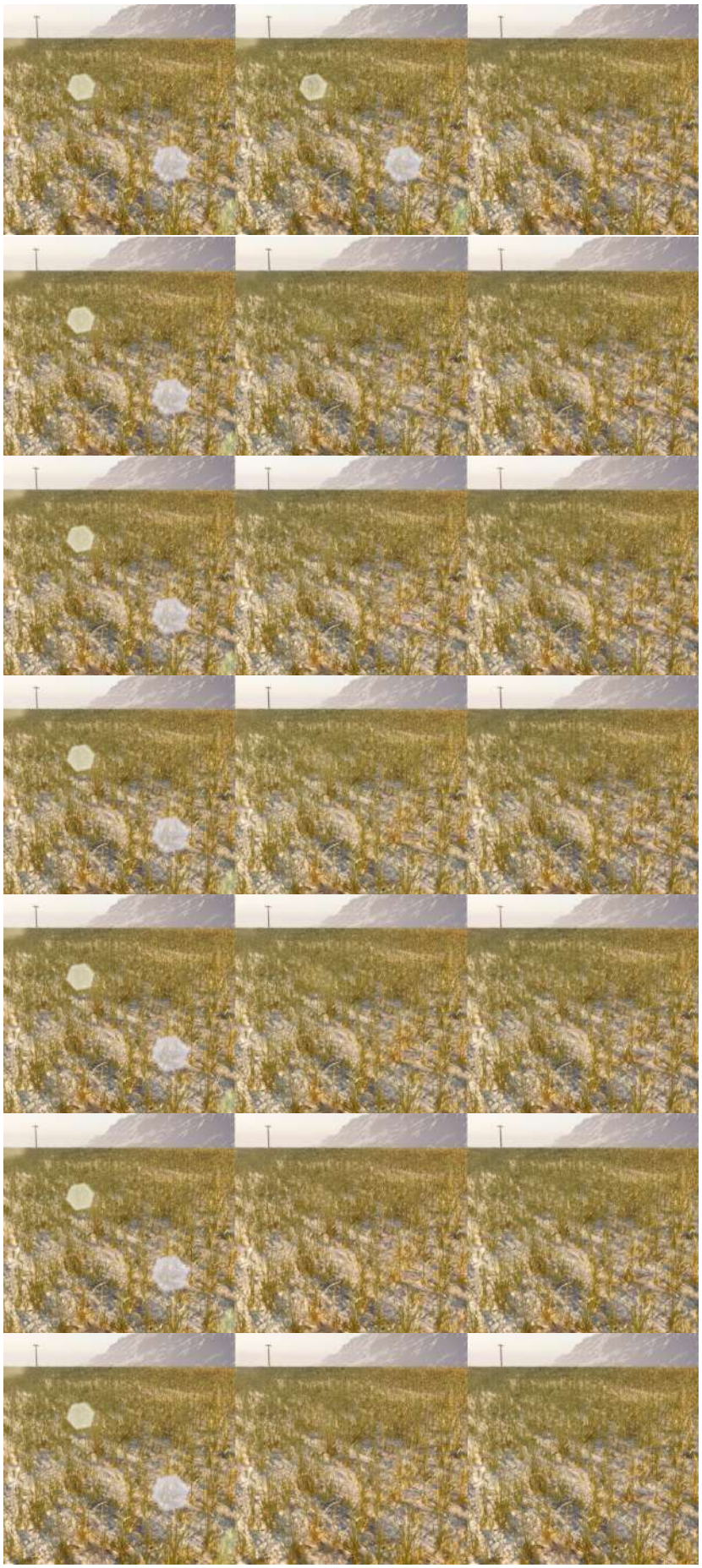


Figure 6.15: Qualitative comparison of flared, restored, and ground truth images for Training 1 to 7. Each row shows the flared image, restored image, and ground truth, respectively.

As observed in the image results above, during the initial training phase (Training 1), the GAN demonstrates a partial ability to identify flare regions but largely fails to remove them effectively. The restored outputs frequently resemble the inputs that were impacted by the flare, which suggests insufficient learning of flare characteristics and a limited capacity for restoration. Both flare detection and removal are significantly improved by the model in later training stages. The GAN accurately localizes the flare-affected regions and begins to reconstruct them with reasonable quality, although some visible artifacts remain in the restored areas. These residual distortions suggest that while the model has learned to suppress flare, it had not yet fully generalized to complex flare patterns or diverse scene content.

The GAN performs noticeably better by the last training cycle. Flare detection is precise, and the restoration process produces high-quality outputs with minimal to no visible artifacts. Interestingly, even areas that have been severely damaged are convincingly reconstructed, suggesting the model has learned both the structure and texture of the underlying scene. These results highlight not only the progressive improvement of the model but also reveal specific conditions—such as flare intensity, camera angle, and vegetation texture—under which the model performs exceptionally well. On the other hand, they also highlight scenarios where performance degrades, pointing to the need for further training using an expanded and more diverse dataset to ensure greater robustness and generalization.

6.3 Performance Discussion

6.3.1 Inference Time

In addition to the quality metrics, the efficiency of the model was evaluated based on its average inference time. The model's speed is crucial, especially for real-time applications where timely predictions are required. As shown in Table 6.3, the model's inference time varied slightly across different training configurations. The average inference time for each configuration ranged from 0.010077 seconds to 0.010355 seconds per sample, which corresponds to an approximate frames per second (FPS ranging from 96.57 to 99.24 FPS). These results indicate that the model is not only effective in restoring images but also capable of operating at a reasonable speed, making it suitable for practical deployment in time-sensitive tasks.

Table 6.3: Inference Time Results for all Training Configurations

Configuration	Inference Time (seconds)	FPS
Training 1 (0028)	0.010318	96.92
Training 2 (0029)	0.010355	96.57
Training 3 (0030)	0.010255	97.51
Training 4 (0035)	0.010323	96.87
Training 5 (0037)	0.010289	97.20
Training 6 (0039)	0.010209	97.95
Training 7 (0041)	0.010077	99.24

Chapter 7

Conclusions

7.1 Synopsis and Final Conclusions

This thesis addresses the challenge of lens flare removal from single RGB images using machine learning techniques, particularly Generative Adversarial Networks (GANs). The study introduces a novel approach that leverages GAN-based architectures with spatial attention mechanisms to efficiently detect and remove flare artifacts in images taken in agricultural landscapes.

The primary outcomes of this research are the development of simulated datasets generated with Unreal Engine to mimic lens flares in a variety of outdoor conditions and the training of the SpA-GAN architecture, originally designed for cloud removal, to restore images contaminated by flares. The results illustrate that the model trained in this way contributes significantly to the restoration of images by uncovering important visual data, e.g. the structure of vegetation that is usually hidden due to flare.

The quantitative evaluation, using metrics such as PSNR and SSIM, confirms the model's strength in image restoration. In particular, the model shows strong performance in retaining structural integrity, as indicated by the high SSIM values in different training setups. In addition, the application of the Green-Red Vegetation Index (GRVI) as a supporting measure has confirmed the model's ability to maintain intact vegetation characteristics, thus verifying its usability in precision farming and ecological observation.

However, the research revealed a few shortcomings. Despite the model's talent in handling this issue, it still faces difficulty when it comes to highly intricate flare patterns, especially in cases of thickly overlaid flare artifacts. Future work might focus on improving the

performance of the model in such exceptional situations and further extending its generalization to cover diverse environmental conditions.

7.2 Future Work

There are several areas where improvements can be made in order to enhance the capabilities of the SpA-GAN in flare removal. Future research could focus on enhancing the model's performance by incorporating additional layers of complexity in the training data, such as varying weather conditions (rain, snow, fog, etc.) and more diverse types of lens flares. This would make the model more adaptable to a wider range of flare scenarios. Another possible research direction is to combine data from multiple sources, such as thermal or LiDAR data, which could help the model to better discern between flare-caused artifacts and actual content of a scene, particularly when the environment is very challenging. Furthermore, future work should include training the model with real data collected from actual environments, as this would provide more accurate results and allow the model to adapt to the natural variability found in real-world scenes, ultimately improving its generalization capabilities. In summary, the present thesis is a step forward in the area of image restoration through the use of GANs and spatial attention mechanisms and opens the door to further advancements in image processing as applied to agricultural and environmental undertakings.

Bibliography

[1] Goodhart Photography. Photo challenge: Photographing sun flare, 2025.

[2] John Terra. What is a generative adversarial network? types, how they work, pros, and cons, 2024.

[3] Minghan Fu, Huan Liu, Yankun Yu, Jun Chen, and Keyan Wang. Dw-gan: A discrete wavelet transform gan for non-homogeneous dehazing. *arXiv preprint arXiv:2104.08911*, 2021.

[4] Rui Qian, Robby T Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. *arXiv preprint arXiv:1711.10098*, 2018.

[5] Heng Pan. Cloud removal for remote sensing imagery via spatial attention generative adversarial network. *arXiv preprint arXiv:2009.13015*, 2020.

[6] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Pearson, 2018.

[7] Eino-Ville Talvala, Andrew Adams, Mark Horowitz, and Marc Levoy. Veiling glare in high dynamic range imaging. *ACM Transactions on Graphics*, 26(3):37, 2007.

[8] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[9] Yuyan Zhou, Ruicheng Kwan, Jinwei Qian, Jiantao Liu, Xin Lin, Chenggang Wang, and Yanqing Liu. Improving lens flare removal with general-purpose pipeline and multiple light sources recovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17777–17786, 2023.

[10] Yicheng Wu, Qiurui He, Tianfan Xue, Rahul Garg, Jiawen Chen, Ashok Veeraraghavan, and Jonathan T Barron. How to train neural networks for flare removal. *arXiv preprint arXiv:2011.12485*, 2021.

[11] IBM. What is machine learning?, 2021.

[12] IBM. What is deep learning?, 2021.

[13] Daniel Dominguez. Understanding inference-time compute, January 2025.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. pages 1125–1134, 2017.

[16] Meng-Hao Guo, Tian-Xing Xu, Jiang-Jiang Liu, Zheng-Ning Liu, Peng-Tao Jiang, Tai-Jiang Mu, Shao-Hua Zhang, Ralph R Martin, Ming-Ming Cheng, and Shi-Min Hu. Attention mechanisms in computer vision: A survey. *Computational Visual Media*, 8(3):331–368, 2022.

[17] R Sudarshana, Sudhir S Hegde, and P Aparna. Uam-net: Robust deepfake detection through hybrid attention into scalable convolutional network. *Expert Systems*, 2025.

[18] Jiawei Li, Zihan Xu, Yiming Zhang, and Lei Wang. Deepfake detection with spatio-temporal consistency and attention, 2025.

[19] Hajar Emami, Majid Moradi Aliabadi, Ming Dong, and Ratna Babu Chinnam. Spa-gan: Spatial attention gan for image-to-image translation. *arXiv preprint arXiv:1908.06616*, 2019.

[20] National Instruments. Peak signal-to-noise ratio as an image quality metric. https://www.ni.com/en/shop/data-acquisition-and-control/add-ons-for-data-acquisition-and-control/what-is-vision-development-module/peak-signal-to-noise-ratio-as-an-image-quality-metric.html?srsltid=AfmBOorcyD-hteLRyl2NaQi9bFGNb4U8I-R-IlwGS6o2mpvuEWhoQdfk.

[21] Abhishek Kumar Pandey. Structural similarity index (ssim). https://medium.com/@akp83540/structural-similarity-index-ssim-c5862bb2b520, 2024.

[22] Abhishek Kumar Pandey. Applicability of green-red vegetation index for remote sensing of vegetation phenology. https://www.researchgate.net/publication/47426815_Applicability_of_Green-Red_Vegetation_Index_for_Remote_Sensing_of_Vegetation_Phenology, 2024.

[23] Epic Games. Unreal engine. https://www.unrealengine.com/en-US, 2025.

[24] Kinza Yasar and Sarah Lewis. What is pytorch? https://www.techtarget.com/searchenterpriseai/definition/PyTorch, 2022.

[25] Fab. Fab.com, 2025.

[26] Fab.com. Wheat grass. https://www.fab.com/listings/9070350f-dea8-40ec-ba57-0777a2c1fb24, 2025.

[27] Fab.com. Soil mud. https://www.fab.com/listings/11eef52b-2c9b-4123-b4de-df266f4ac044, 2025.

[28] Fab.com. Grass. https://www.fab.com/listings/d87922e7-3131-49b3-bcd7-ae05f3731721, 2025.

[29] Fab.com. Round hay bale. https://www.fab.com/listings/fc0388e8-5443-4e54-b618-65790804779b, 2025.

[30] Fab.com. Sty and telegraph pole. https://www.fab.com/listings/3a19868a-68d4-47cb-b5a0-580161377c54, 2025.

[31] Fab.com. Trees. https://www.fab.com/listings/f95c6c4b-9ff3-4315-ae74-9013c212b136, 2025.

[32] Fab.com. Grassy mountains. https://www.fab.com/listings/b790dd9d-4ab0-4dc8-94f0-3811121b80b0, 2025.

[33] Fab.com. Warehouse building. https://www.fab.com/listings/1876bb93-5a5f-486e-9ba9-82f299b6b05e, 2025.