

SINGLE IMAGE LENS FLARE REMOVAL USING DEEP LEARNING TECHNIQUES

Submitted by

AMULYA PRASHANT KARKAL

180929142

ARYAN JAMES PHILIP

180929144

Under the guidance of

Dr ASHA C S

Associate Professor

Department of Mechatronics

MIT Manipal

in partial fulfilment of the requirements for the award of the degree of

**BACHELOR OF TECHNOLOGY
IN
MECHATRONICS**



**DEPARTMENT OF MECHATRONICS
MANIPAL INSTITUTE OF TECHNOLOGY**
(A Constituent of Manipal Academy of Higher Education)
MANIPAL - 576104, KARNATAKA, INDIA
June 2022



MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

DEPARTMENT OF MECHATRONICS

Manipal
June 17, 2022

CERTIFICATE

This is to certify that the project titled **SINGLE IMAGE LENS FLARE REMOVAL USING DEEP LEARNING TECHNIQUES** is a record of the bonafide work done by **AMULYA PRASHANT KARKAL (180929142)** and **ARYAN JAMES PHILIP (180929144)** submitted in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology (B.Tech.) in MECHATRONICS of Manipal Institute of Technology, Manipal, Karnataka, (A Constituent unit of Manipal Academy of Higher Education), during the academic year 2021-2022.

Dr ASHA C S

Associate Professor

Dept. of Mechatronics

MIT Manipal

Dr D.V. KAMATH

Professor and HOD

Dept. of Mechatronics

MIT Manipal

ACKNOWLEDGEMENTS

To begin, we would like to express our gratitude to everyone who helped, endorsed, and motivated us throughout the project to ensure its success.

Dr Asha C S, our project advisor, is to be thanked for guiding us through the intricacies of lens flare removal, explaining the methodology to us that has been previously developed and also providing us with the resources to make this project a success. Her optimism, persistence, valuable suggestions, necessary details and practical advice have been invaluable to us throughout our research and writing of this report. We are also grateful to our HOD, Dr. D.V. Kamath, for giving us the opportunity to work on this project.

Finally, we appreciate the support from our families and friends for having provided us with the support and encouragement that we needed to complete this project successfully on a regular basis.

ABSTRACT

When it comes to collecting a dataset for image processing, OpenCV, or object detection, photographs are extremely important. Images or photographs are also important in a variety of fields, including cinematography, advertising, marketing, portfolio creation, and much more. Any aberration will obstruct the creation of high-quality photographs. Flares are a specific type of aberration that we are attempting to address in this project. When it comes to flare removal, there isn't a lot of research done. The goal is to devise a method to remove flares without causing damage to the original image.

A deep convolution neural network is proposed along with a well-constructed loss function to create a model that can be used in the flare removal process. However, this is only possible with a good dataset. The dataset used was created artificially and consists of readily available images of just flares superimposed on ground truth images. These images are saved in pairs of ground truth and synthetically generated images, which is the dataset used for training. After a large number of epochs, it is discovered that the model outperforms the ones that were previously used to remove flares.

LIST OF FIGURES

1.1	Different types of aberrations in images	2
1.2	Lens Elements	3
1.3	Flare inducing light source	3
3.1	Figure of a generalized representation of a GAN.G and D represent the generator and the discriminator. θ_g and θ_d define the parameters of the generator and discriminator.	9
3.2	The architecture of a CGAN model [16]	11
3.3	U-net (Generator) architecture [17]	11
3.4	PatchGAN (Discriminator) architecture [18]	12
3.5	Images translated from source to target [19]	12
3.6	Images are captured at different angles using a strong light source. The camera is mounted on a swivel base that produces different incident angles [9]	13
3.7	The original synthetic image, the flare preset, and the final combined image.	13
3.8	ResNet18 Architecture [20]	14
3.9	AlexNet Architecture [21]	15
3.10	SqueezeNet Architecture [22]	15
3.11	VGG16 Architecture [23]	16
3.12	MobileNet V2 architecture [24]	16
3.13	The flare features extracted from the dataset	17
3.14	The cGAN architecture used in our model	18
3.15	A U-Net architecture with batch normalization [25]	19
3.16	Proposed PatchGAN model detects between generated and ground truth images.	19
3.17	Generator and Descriptor Losses of our model	21
5.1	Color channel visualization of an image in the dataset	26
5.2	Results of the Flare Removal Algorithm	27
5.3	Architecture of FFAnet [11]	29
5.4	Results of FFANet architecture	29
5.5	CycleGAN architecture	30
5.6	Results of CycleGANet Architecture	30
5.7	HINET architecture [27]	31
5.8	Results of HINET architecture	32
5.9	NAFnet architecture [28]	32
5.10	Results of NAFnet architecture	33
5.11	GMAN architecture [10]	33
5.12	Results of GMAN architecture	34

5.13	Result Analysis	35
5.14	Failure case when the flare and light source are disjointed	36
5.15	Failure case when the light source is too bright	36

LIST OF TABLES

2.1	Comparison between previous methods used for flare removal	7
3.1	Using a metrics system, all the previous research results with the method used to achieve it.	23
3.2	Parametric Data of Flare Removal using two Generator architectures.	23
5.1	Result analysis of Flare classification using SOTA deep learning approaches	27
5.2	PSNR and SSIM result analysis on two different test datasets	28
5.3	Quantitative analysis of Flare Removal using SOTA deep learning approaches	34

LIST OF ACRONYMS

PSNR Peak Signal to Noise Ratio

SSIM Structural Similarity Index Measure

GMAN Graph Multi-Attention Network

CNN Convolutional Neural Network

FFA-Net Feature Fusion Attention Network

FA Feature Attention

DCP Dark Channel Prior

HSV Hue, Saturation, and Value

GAN Generative Adversarial Network

CGAN Conditional Generative Adversarial Network

FPGA Field Programmable Gate Array

ReLU Rectified Linear Unit

VAE Variational Autoencoder

FC Fully Connected

SOTA State of the Art

TABLE OF CONTENTS

Acknowledgements	i
Abstract	ii
List of Figures	iv
List of Tables	v
List of Acronyms	vi
Chapter 1 Introduction	1
1.1 Different types of image distortion and their cause	1
1.2 Causes of Lens Flare	2
1.3 Ways to avoid Lens Flare	3
Chapter 2 Literature Review	4
2.1 Conventional Approaches	4
2.2 Problem Definition	8
2.3 Objectives	8
Chapter 3 Methodology	9
3.1 Theoretical Background	9
3.2 Creating a Synthetic Dataset	13
3.3 Training a Feature Extractor	14
3.4 Flare Removal Algorithm	17
3.5 Training Setup	22
3.6 Comparison of state-of-the-art models	23
Chapter 4 Contribution of Each Student	24
Chapter 5 Results and Discussion	26
5.1 Dataset	26
5.2 Flare Detection	26
5.3 Flare Removal Algorithm	27
5.4 Comparison of Flare Removal Algorithms	28
5.5 Result Analysis and Discussion	34
Chapter 6 Conclusion and Future Scope	37
6.1 Conclusion	37
6.2 Future Scope	37

References	40
Annexures	41
Project Details	49
Plagiarism Check	50

CHAPTER 1

Introduction

The human brain processes images better than text to interpret information faster and retain more data. That is precisely why almost every other industry uses images for communication. News outlets attach pictures to clarify the incident they write about, advertisements for products use enhanced images to attract consumers' attention, and live streaming of sports events or scientific events all depend on the image quality.

Nevertheless, disturbances and aberrations may occur at any image generation and processing stage. Any form of noise or distortion in an image ruins the overall quality and usability of the image and is mainly seen as a hindrance. To prevent one from taking bad-quality images, one must understand the various distortions that can prevent one from taking good-quality images.

Lens flares are common image artefacts amongst several other forms of artefacts. Due to its diversity in appearance and the wide variety of patterns in lens flares, it becomes challenging to remove them from images. Hence, one must understand how lens flares occur in images. After careful analysis of such image artefacts, tools and frameworks have to be developed that remove them and improve the overall image quality.

1.1 Different types of image distortion and their cause

Haze – Scattering atmospheric particles degrade images captured in hazy or foggy weather conditions, reducing contrast, changing colour, and making object features challenging to identify by human vision and some outdoor computer vision systems.

White Noise – Image noise is the random variation of brightness or colour information in collected photographs. The picture signal is deteriorating due to external factors. According to multiplication noise images, the brighter the area, the noisier it is. However, it is mainly additive. Images are transmitted electronically from one location to another. White noise has several causes. While taking an image, detect heat with a variable ISO factor that fluctuates with the camera's ability to absorb light.

Raining– As a complex atmospheric process, rain can degrade visibility in a variety of ways, depending on a variety of environmental factors such as raindrop size, rain density, and wind velocity. When photographing a rainy scene, many camera settings, such as exposure time, depth of field, and resolution, influence the visual effects of rain on the digital image. Most existing deraining projects use a single rain model (usually a rain streak), which may have oversimplified the problem. Existing rain models are divided into three categories in the litera-

ture: rain streak, raindrop, and rain and mist.

Flares - A lens flare occurs when bright light enters the camera lens, hits the sensor, and flares outwards. In this process, the intense light source may go through multiple reflections or scatter within the lens due to dust or scratches. While lens flare is used as a stylistic artifact in several images, it can also easily distract the audience from the intended target subject. It can also wash out or damage an image when done incorrectly or unintentionally.

In this project, a comprehensive study is done on the artefact of lens flare and how it affects the quality of an image. A comparison is made with the existing projects that propose solutions to remove lens flares from images. Finally, after a thorough understanding of the above problem, a deep-learning framework is proposed that removes lens flares from images.

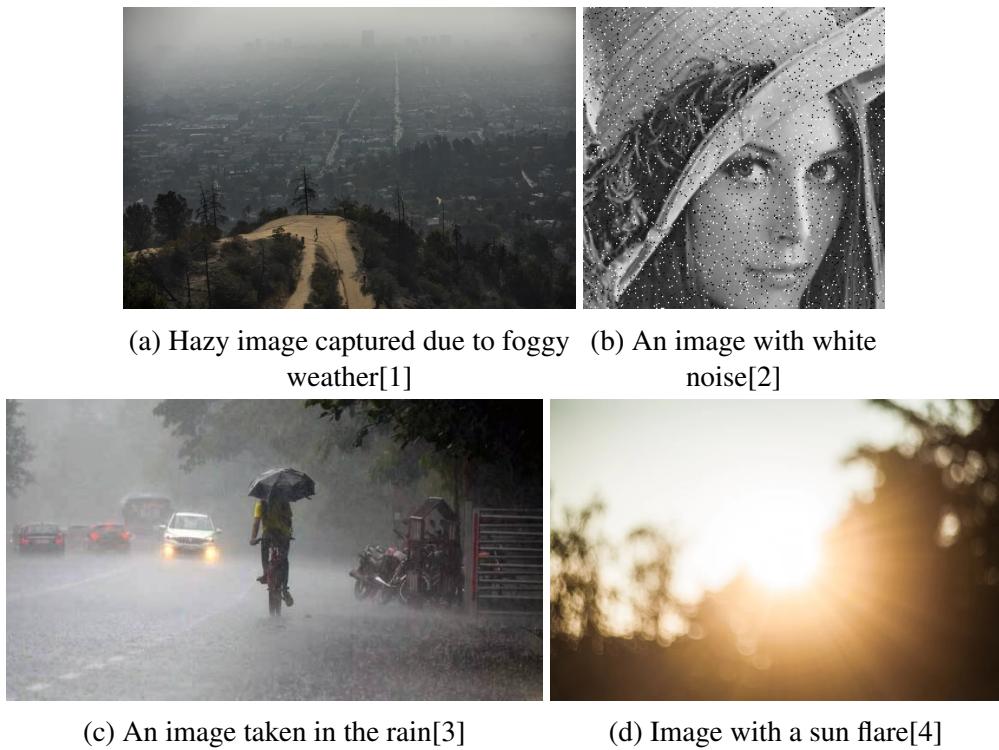


Fig. 1.1: Different types of aberrations in images

1.2 Causes of Lens Flare

The most common source of lens flares is when non-image light refracts many times and does not follow the planned path. Instead, the light reflects numerous times internally on lens elements and bounces back and forth before reaching the film or digital sensor. Anti-reflective coatings are commonly used on elements to reduce flare, but no multi-element lens completely removes it. When a small amount of light from a source reflects, the reflected light appears as a flare in some areas. The intensity is comparable to reflected light until it becomes comparable to refracted light in power. Light reflected off the inside causes flare, which appears as polygonal forms.

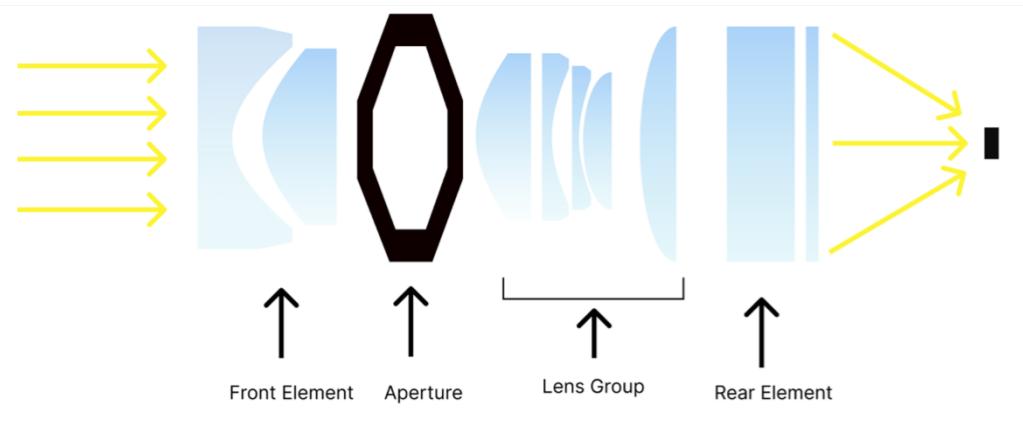


Fig. 1.2: Lens Elements

Internal reflections can theoretically cause a flare, but observing one often necessitates the use of very bright light sources. The sun, artificial lighting, and even a full moon are flare-inducing light sources. Even if there are no bright light sources in the shot, stray light can enter through the front element. Outside-of-angle-of-view light usually does not contribute to the final image, but it can take an unexpected path to the film/sensor if it reflects.

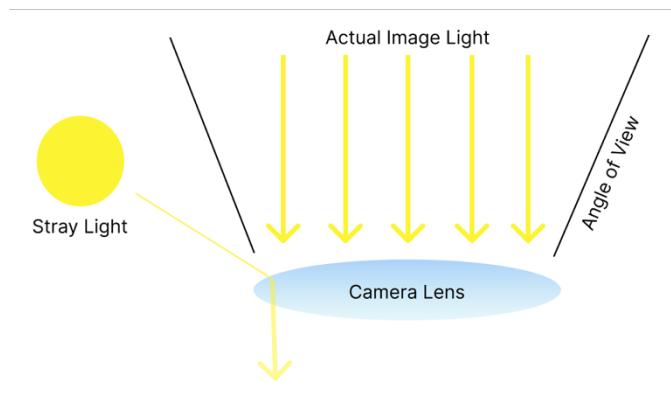


Fig. 1.3: Flare inducing light source

1.3 Ways to avoid Lens Flare

To avoid lens flare, you must first understand how it occurs. Lens flare occurs when bright light strikes your lens directly or when the light source is at a precise angle to the lens. The lens flare will be more noticeable the closer the light source is to the edge of your frame. To reduce lens flare, follow these simple steps: Shoot with the light source in the back of your head. To block the fit, use a lens hood. To avoid harsh light, schedule your shoots during golden hour. Lens flare may occur. There's no direct line of sight between the front lens and the light source to avoid lens flare.

CHAPTER 2

Literature Review

Millions of images are taken daily, and the uses extend to various fields. Images are crucial for making science observations, data collection for computer vision, or memory capture. Nevertheless, the images may develop flares due to factors such as a vital source of light or a cloudy camera lens. These flares reduce the quality of the photos. They obstruct one from making accurate observations and can be seen as an inconvenience. With rigorous analysis and experiments, a framework is developed to mitigate the effect of these flares and achieve a better quality photo. There are various methods used to accomplish this.

This section will discuss the existing research done to remove different abnormalities from images—irregularities such as noise, rain streaks, haze, and flares. Various the conventional methods for many works will be mentioned and further be discussed. This includes different deep learning models that have attained practical relevance in PSNR and SSIM. Finally, there is an examination of the various datasets, measurements, and losses that have benefited this research.

2.1 Conventional Approaches

Image flare detection and removal are fundamental issues that need to be tackled. Flare artefacts have been measured, detected, and removed using various methods, including hardware and software. Most hardware solutions focus on improving the camera’s optical elements to eliminate flares.

Criminisi et al. [5] He and others proposed a method for filling a specific region in a photo when it is missing or removed. They start by locating the missing pixel patch. They then determine the patch’s priority by calculating which pixel should go in which patch region. They also make certain that the texture and structure information is consistent with the pixels at the edges. They calculate the confidence value and proceed after a specific region has been filled. The confidence values are updated each time these steps are completed. As a result, the innermost pixel’s confidence values are close to 0.

Lee et al. [6] proposed a real-time method for rendering lens flare effects. Expensive equipment was previously used to trace the ray or derive complex polynomial expressions. The approach that is proposed is derived from the first-order approximation of ray transfer in an optical system as an alternative. A matrix is created that directly maps lens flare-producing light rays to the sensor. The resulting approach is simple to implement on standard off-the-shelf graphics hardware and produces physically plausible images at high framerates.

Victoria et al. [7] proposed a reliable computational method for automatically detecting and removing flare spot artefacts. There were three parts to the contribution. The first step was a characterization based on intrinsic properties that a flare spot is likely to satisfy. Second, they devised a new method for detecting flare spots among a large number of photos. The final step included a plan to determine the flare region in a given image. These artefacts are removed using exemplar-based inpainting after the detection is complete.

For single-image flare removal, Qiao et al. [8] propose a deep framework with light source-aware guidance. The first step is to distinguish between light source regions and flared regions. The flare artefacts are then removed using the light source-aware guidance as a guide. They could remove various types of flares from the images by understanding the relationships between the two steps mentioned above. They also propose the first unpaired flare removal dataset to obtain more diverse examples and avoid manual annotations instead of using difficult-to-collect paired training data.

Wu et al. [9] suggested creating pairs with their ground truth images using synthetically generated flared images. This dataset aids the training model's detection of flares. Their CNN model successfully detects the light source as well as the flare, and removes both from the flared image, which is very undesirable. As a result, adding the light source back into the resulting image is their final step in restoring the ground truth image. Their PSNR is 25.5 and their SSIM is 0.85.

2.1.1 Dehazing, Deraining, and Denoising

Liu et al. [10] propose in their paper a generative method to remove haze from images. Haze prevents the image from being viewed with proper clarity. They make use of a GMAN which detects the haze and removes it from the input images. GMAN makes use its fully convoluted network and recovers the original image without making any reference to the scattering model of the atmosphere. PSNR = 28.19 and SSIM = 0.9638

In their paper, Qin et al. [11] claim that using an FFA-Net successfully removes haze from images. Three major components make up their architecture. To begin, it uses an FA module to add different weights to different haze pixels. This is significant because the haze distribution across the image cannot be expected to be uniform. This provides CNN with additional data. Then they run a feature extraction module to ignore features with low weights compared to the effective features, rendering them useless for network training. Finally, these features' weights continue to learn from the FA and give more weight to the relevant features. This aids the network's deeper layers in retaining useful information.

Chen et al. [12] propose a new method for restoring images with low visibility. They created an HINet that improves other image restoration blocks such as the ones used for dehazing

and denoising. The HINet network is a highly convulsed network with two U-Nets serving as subnetworks. The PSNR and SSIM values of this block have increased significantly.

He et al. [13] propose a dark channel prior (DCP), which is made up of minimum intensities over three tracks in a given neighbourhood. Because the haze component is generally white, adding equal powers to all the colour channels, the dark track for hazy images is brighter than the dark channel for non-haze ideas. He et al. [13] compares 0.1 percent of the brightest pixels in the dark channel, assuming that the hazing is homogeneous. Atmospheric light is the pixel in the input image with the highest value. In subsequent work, Chen et al. [12] demonstrated that using the guided filter instead of soft matting improves the DCP algorithm's performance.

Zhu et al. [14] are based on the colour attention prior based approach. The paper observes that haze density increases with the increase in depth. Therefore, depth is positively correlated to the concentration of haze, which in turn depends upon the difference between the value and saturation of the pixel in Hue Saturation Value (HSV) space. The paper proposes a linear model for modelling the scene depth.

While it is common knowledge that converting synthetic to authentic images necessitates careful consideration of camera sensor noise properties, other aspects of the image processing pipeline (such as gain, colour correction, and tone mapping) are frequently overlooked, despite their significant impact on how raw measurements are transformed into finished images. Author have proposed inverting each step of an image processing pipeline to generate realistic, raw sensor measurements from widely available Internet photos. They simulate the necessary components of an image processing pipeline when evaluating our loss function, allowing training to be aware of the relevant photometric processing after denoising.

Table 2.1: Comparison between previous methods used for flare removal

Year	Article	Metrics	Methodology	Observations
2004	"Region filling and object removal by exemplar-based image inpainting." [5]	No metric system is used to compare the results. The images are placed side by side, and observations are made.	The model identifies patches, computes the path priorities, propagates texture and structure information and updates confidence values	The target region in the image is grown with sharp linear structures. There is no single structural element that dominates the others. The naturally decaying confidence values achieve this balance
2019	"Automatic Flare Spot Artifact Detection and Removal in Photographs". [7]	Dice similarity coefficient. It is a metric used to calculate the similarity between two sets	Characterizations of flares followed by the determination of the flare region. These flares are removed by exemplar-based inpainting	A bar graph with the values of dice coefficient found by using the Automatic Lens Flare Removal algorithm [15] and their algorithm.
2021	"How to Train Neural Networks for Flare Removal." [9]	PSNR SSIM	A synthetic dataset is created which is trained using a CNN model. The final step includes externally adding the source back into the image	PSNR = 25.55, SSIM = 0.85
2022	"Light Source Guided Single-Image Flare Removal from Unpaired Data" [8]	PSNR SSIM	Unpaired dataset is used. The light source and flares are detected and the flare removal module removes the flares. Two different discriminators are used to train flare images and flare free images	The PSNR and SSIM values are found compared with Wu et al. [9]. They used 48 images for comparison and returned the values as follows: PSNR – 21.57 SSIM – 0.812

2.2 Problem Definition

As stated earlier, this project is taken up to combat the visibility issues faced due to flares in images. It is inconvenient for people and machines that require clear images as their input. Clear images have significant advantages as clear photos help create a training dataset for accurate results, and object detection requires clear images to avoid accidents. It helps in applications such as local path planning. To solve this problem, this report proposes a model to train synthetically generated flared images using a GAN to remove the flares to match with the ground truth image. This will have various advantages as the model is used on flared images, and it would be able to remove the flare and create the original photo without having a ground truth image.

2.3 Objectives

- Appreciate the usage of images in our daily lives and in the professional world. Analysing the different ways to solve the problem definition by researching all the previous methods that have been used to achieve it.
- We create a dataset of standard images with different flares synthetically added to them. Arrange the dataset so that the ground truth images are in pairs with the synthetically generated ones for better comparison purposes. They are creating a GAN model to train the dataset.
- We are using various methods to calculate the error for comparing our model's accuracy with the previous being used models.

CHAPTER 3

Methodology

3.1 Theoretical Background

3.1.1 What are Generative Adversarial Networks?

The advent of deep learning in the late 1990s and early 2000s by Yann Le Cun and Geoff Hinton brought about a new wave of interest in a subject that was thought to be dead. Their research on classifying handwritten digits using neural networks gave rise to the concepts widely used today in medicine, self-driving vehicles, and many other fields. However, this temporary hype died down quickly until companies like NVIDIA started developing GPUs that could take advantage of large datasets and complex models having numerous matrix multiplications. This also gave rise to frameworks such as Tensorflow and Pytorch, making models and loading data much more effortless. However, with all the new advances, infinite possibilities of AI was stagnated until a new concept was presented in 2014, called Generative Adversarial Networks (GANs).

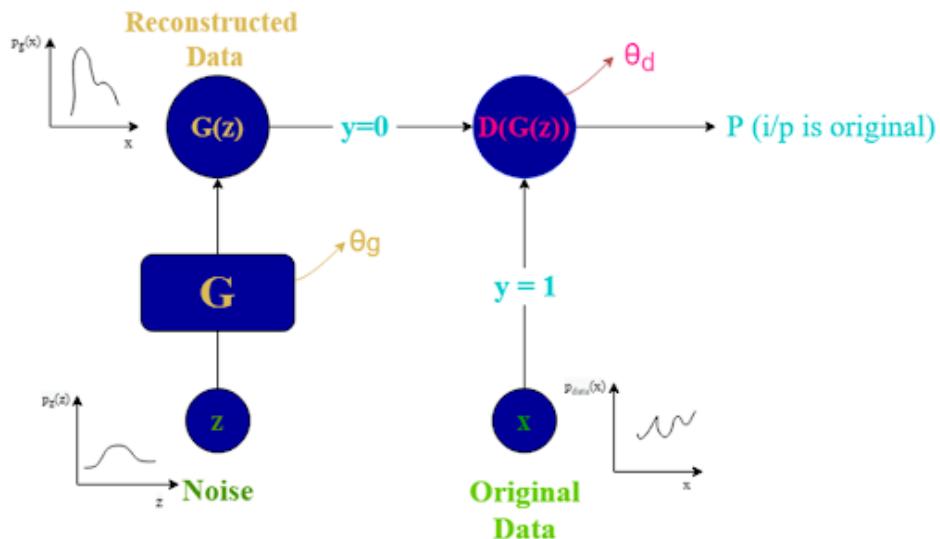


Fig. 3.1: Figure of a generalized representation of a GAN. G and D represent the generator and the discriminator. θ_g and θ_d define the parameters of the generator and discriminator.

As the name suggests, Generative Adversarial Networks tries to generate something new from a random noise vector given as an input to the system. This is where the generative part comes in to produce fake samples. The next part is the Adversarial network, wherein the generator and discriminator play a cat and mouse game; the generator tries to fool the discriminator by producing samples as close to the original one, and the discriminator tries to produce a binary output to detect whether the sample is original or fake. Now, where does the learning part of this network come in? The generator initially would produce poor quality samples, and the discriminator would easily be able to detect the fake samples and would generate a loss; this loss is then fed back into the generator to produce better samples to fool the discriminator. Eventually, after many epochs, the generator starts to produce samples as good as the ground truth that the discriminator cannot distinguish between the original and fake samples.

$$V(G, D) = \min_G \max_D E_x \sim P_{data}(x) [\log D(x)] + E_z \sim P_z(z) [1 - \log D(G(z))] \quad (3.1)$$

The above equation represents the objective function of GAN.

D is the discriminator

G is the generator

θ_d is the weights and biases of the discriminator

θ_g is the weights and biases of the generator

$P_z(z)$ is the input latent vector

$P_{data}(x)$ is the original data distribution

$P_g(x)$ is the generated data distribution

$V(G, D)$ is the objective function of the GAN

3.1.2 Image to Image Translation

Recent advances in GANs have brought many new SOTA architectures. Pix2Pix is a novel method to transfer images from the source domain to the target domain, provided that the source-target image pairs are present. Here, the concept of conditional GAN is used, wherein the conditional probabilities are calculated in the loss function rather than the marginal probabilities. This helps the generator better understand our source domain and helps in better training.

The generator is a U-net, an encoder-decoder model that reduces the spatial dimensions of the source image to a smaller vector space containing particular features of the source image. These features are high-level features like texture, corners, etc. However, encoding results in

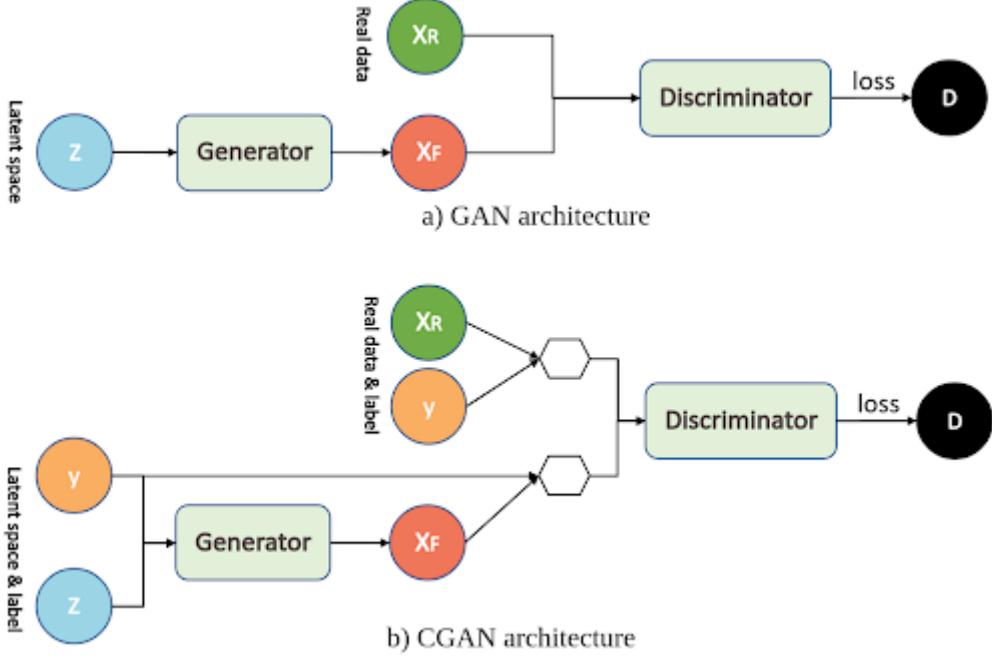


Fig. 3.2: The architecture of a CGAN model [16]

losing a lot of low-level features in the earlier convolutional layers. To rectify this, the outputs of the encoder convolutional layers are used and fed into the corresponding decoder layers. The decoder works to upscale the encoded image into a new image having features similar to the target image. It uses transposed convolutions or deconvolutions to upscale the latent feature vector space.

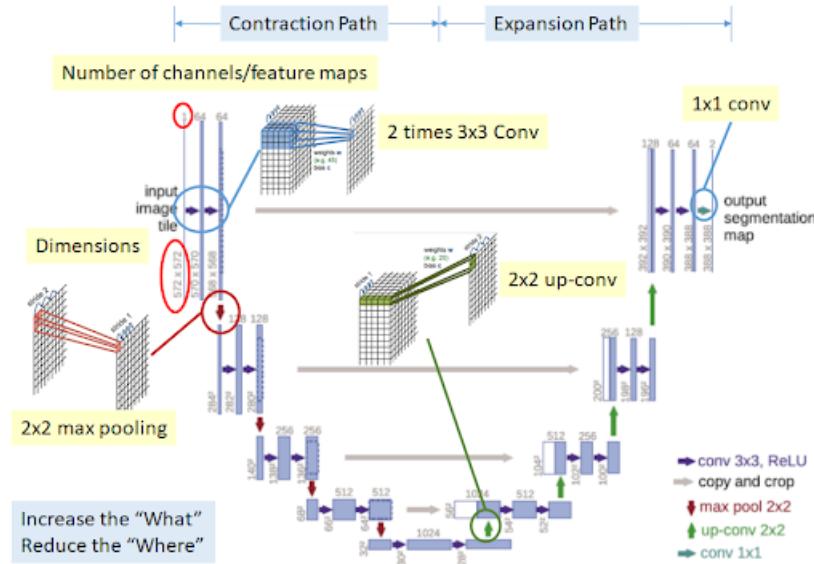


Fig. 3.3: U-net (Generator) architecture [17]

The PatchGAN architecture is very different from the original GAN discriminator, wherein it considers small patches of the image to be classified. The main difference between a PatchGAN and a regular GAN discriminator is that - the standard GAN maps an input image to a

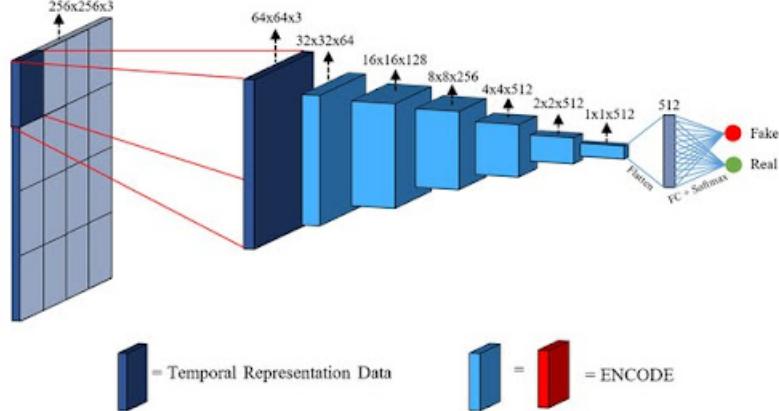


Fig. 3.4: PatchGAN (Discriminator) architecture [18]

single scalar output in the range of [0,1], indicating the probability of the image being real or fake, while PatchGAN provides a matrix as the output with each entry signifying whether its corresponding patch is real or fake. In PatchGAN, given, for example, an image of size 256x256, the PatchGAN maps from that 256x256 to an NxN Matrix of outputs X, where each X_{ij} of that NxN Matrix signifies whether the patch ij (in X) in the image is real or fake. So each of these X_{ij} values (a single scalar value) is a probability for the likelihood that a patch in the input image is real.

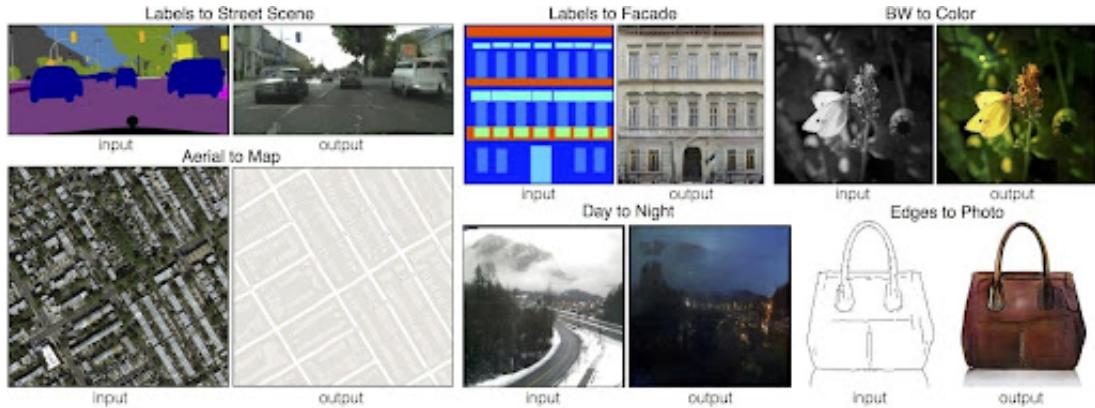


Fig. 3.5: Images translated from source to target [19]

3.2 Creating a Synthetic Dataset

The difficulty of image-to-image translation lies with the dataset. High-quality image pairs consisting of ground truth and flared images are required. Since there aren't any image processing techniques to extract flares from photos, high-quality flares in a controlled environment have to be collected.

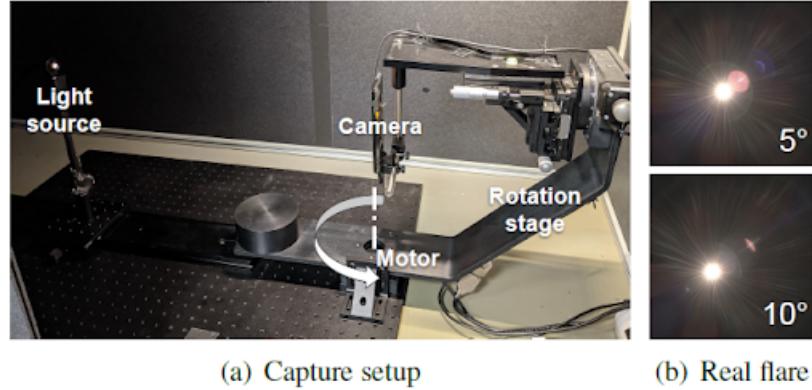


Fig. 3.6: Images are captured at different angles using a strong light source. The camera is mounted on a swivel base that produces different incident angles [9]

Flares collected in this manner require high-quality cameras and a strong light source in an isolated environment, which is time-consuming and expensive. The other method is to take flare pre-sets captured from space missions and merge them with a synthetic image taken from the internet. This flare pre-sets readily available, and with a few image augmentation techniques, more flare pre-sets can be generated. The other advantage is that a wider variety of flares can be used, and our model generalizes better.

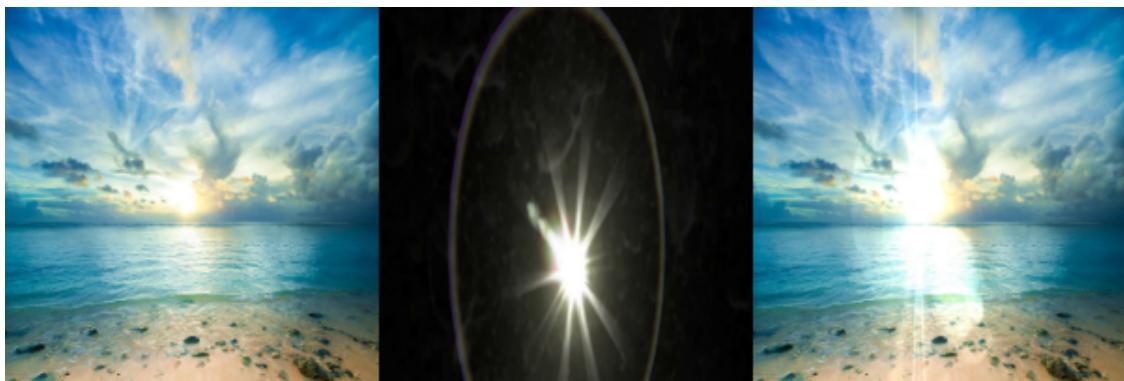


Fig. 3.7: The original synthetic image, the flare preset, and the final combined image.

These images generated are synthetic and various augmentation techniques are applied to increase the size of the dataset. The flare-free images are sourced from the Flickr 24K dataset and only 11388 images are chosen for training and a further 2000 images are chosen for validation and testing. Open CV and PIL libraries are used to create image pairs. The flares were

obtained by extracting frames of an open-source video. A weighted sum of ground truth and the raw flare image was taken for the paired flare image. Image processing techniques are used to accurately crop the flare and superimpose it on the ground-truth image.

3.3 Training a Feature Extractor

Solar flares are usually associated with their corresponding light source, however, in some cases, the solar flares can be spatially separated from their light sources and subtle flare artifacts can get lost during I2I. To avoid this, it was decided to use pre-trained networks as a feature extractor to help localize the region of interest of the corresponding flare artifact and not just the light source. Many pre-trained networks were tested but ended up choosing ResNet 18.

Five SOTA deep-neural networks were tested:

- Resnet18
- AlexNet
- SqueezeNet
- VGGNet
- MobileNetv2

ResNet18: Resnets reformulated the layers of a neural network to learn residual functions concerning input layers. The web is easier to optimize and increases the gain considerably. The original work used the Imagenet dataset with a depth of up to 152 layers which is more profound than VGG Nets. The deep representation of the network helped to increase the accuracy to an extent and secured low error on the ImageNet dataset. ResNet18 is a build using micro blocks. Each microarchitecture consists of convolutional layers, pooling layers, etc. The network is trained using SCG through the use of residual modules.

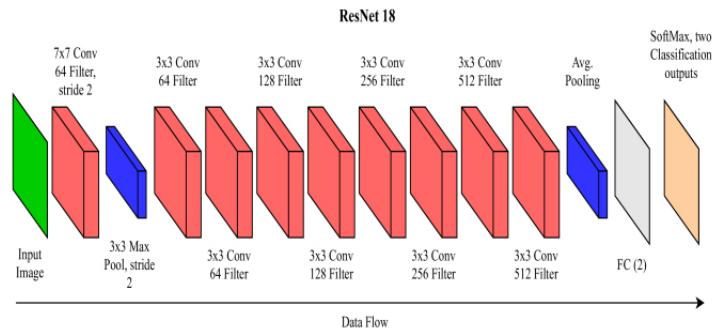
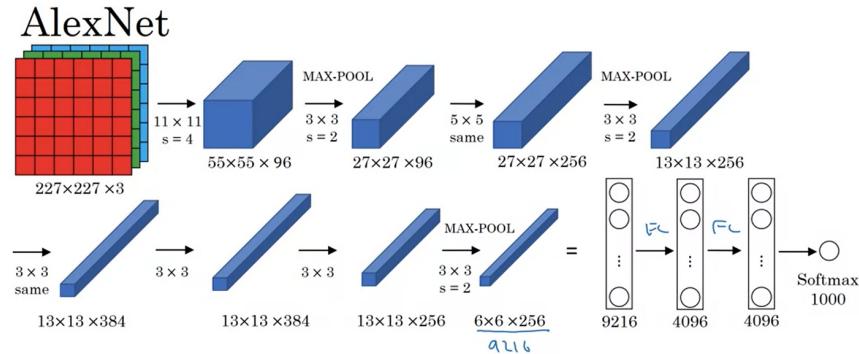


Fig. 3.8: ResNet18 Architecture [20]

(Green) Input Images, (Pink)Convolution Layer, (Blue) Pooling Layer, (Gray) Fully-Connected Layer (Apricot) SoftMax.

AlexNet: Alex et al. proposed a method based on a neural network to classify millions of high-resolution images having 1000 classes. This method has shown significantly improved classification compared to conventional hand-picked feature-based methods. The network consists of five layers with 60 million parameters, max-pooling layers, and three fully connected layers, followed by a softmax layer. The architecture comprises eight learned layers, five convolutional, and three fully connected layers. ReLU is adopted to prevent saturation and applied after each convolutional and fully connected layer. Pooling layers consist of a grid of units spaced 's' pixels apart. The output of the last layer uses binary cross-entropy to classify the image as are or not.



[Krizhevsky et al., 2012. ImageNet classification with deep convolutional neural networks]

Andrew Ng

Fig. 3.9: AlexNet Architecture [21]

SqueezeNet: SqueezeNet proposed CNN architecture with several advantages in which deployment of models on FPGA and other hardware help to dump for autonomous cars. It achieves the AlexNet accuracy with fewer parameters. In addition, it is also possible to reduce the size to less than 50MB. The architecture begins with convolution layers, eight fire layers, and fully connected layers.

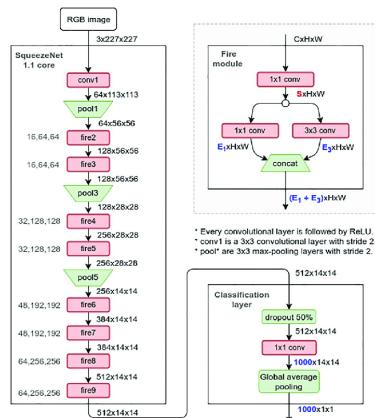


Fig. 3.10: SqueezeNet Architecture [22]

VGGNet: VGGNet increases the depth using small 3×3 filters in the convolutional layers, improving the performance by increasing the depth to 160-19 weight layers. It also used 1×1 convolution filters followed by the max pool, conv, and fully connected layers.

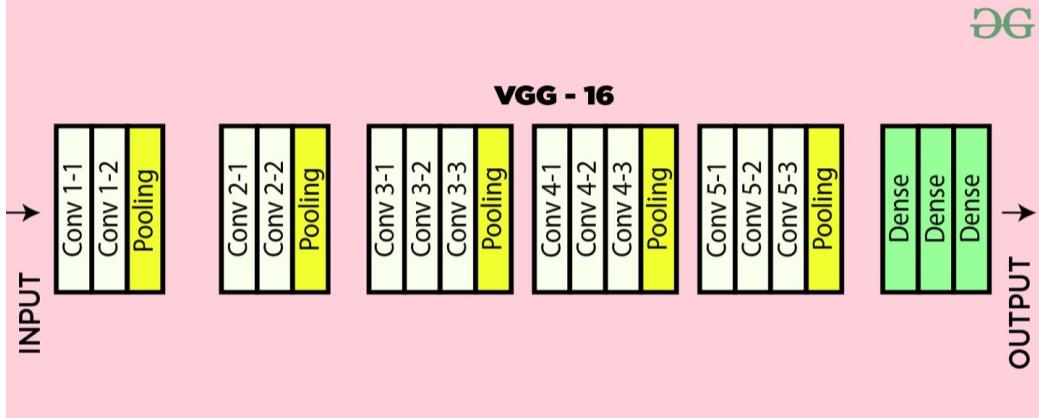


Fig. 3.11: VGG16 Architecture [23]

MobileNetV2: Mark Sandler includes a new mobile architecture to detect objects. The work is based on the inverted residual structure in which shortcut connections are established between the bottleneck layers. The layers in between are lightweight depthwise convolutions to filter the features. It tries to remove non-linearities in the thin layers to retain the symbolic power. This approach also decouples the input and output domains. The model architecture consists of convolutional layers, ReLU, and fully connected layers with 32 filters.

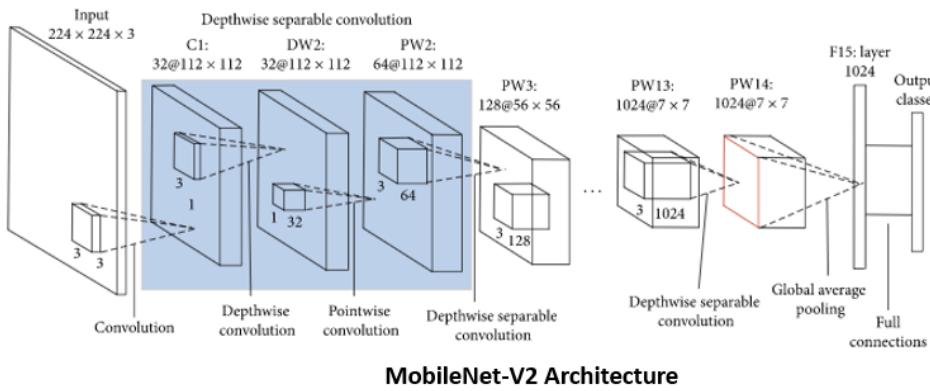


Fig. 3.12: MobileNet V2 architecture [24]

These architectures were used with three convolutional layers to visualize the extracted features. The fully connected layers were removed, which helps to transfer more data from the extracted features.

A flare detection algorithm was run to decide the best feature extractor. However, this problem

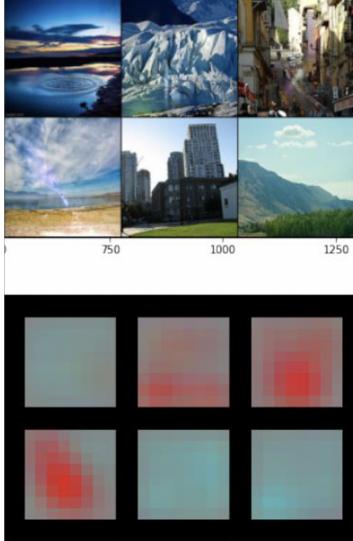


Fig. 3.13: The flare features extracted from the dataset

is quite complex since there are many flare types, and there is a considerable class imbalance. Undersampling was used to achieve the best accuracy on the dataset. From the above feature maps, it is clear that it provides us with spatial information about the flares and helps in our original goal of flare removal. These feature maps will also become helpful during post-processing steps to modify the generated image to its color, saturation, hue, etc.

3.4 Flare Removal Algorithm

Flare removal is quite a challenging and complex problem to tackle. Previous image processing techniques have fallen short in removing flares due to the sheer number of flare varieties as well as their shape and nature. Also, naturally occurring flares are quite distinct in some aspects from the flares generated in a lab. Since flare pre-sets are used in our datasets, our model generalizes well to synthetic flares but performs rather average on natural flares.

To develop our flare removal algorithm, a CGAN approach is used, wherein a U-Net was used as a generator, and Patch GAN was used as a discriminator. Several other generators, including the Attention U-Net and VAEs were tested. While other generators on the dataset provided comparable results, it noticed a problem with mode collapse during training. To keep a system of checks, we use a U-Net with skip connections. An input image of 256x256x3 is used, using batch normalization instead of instance normalization.

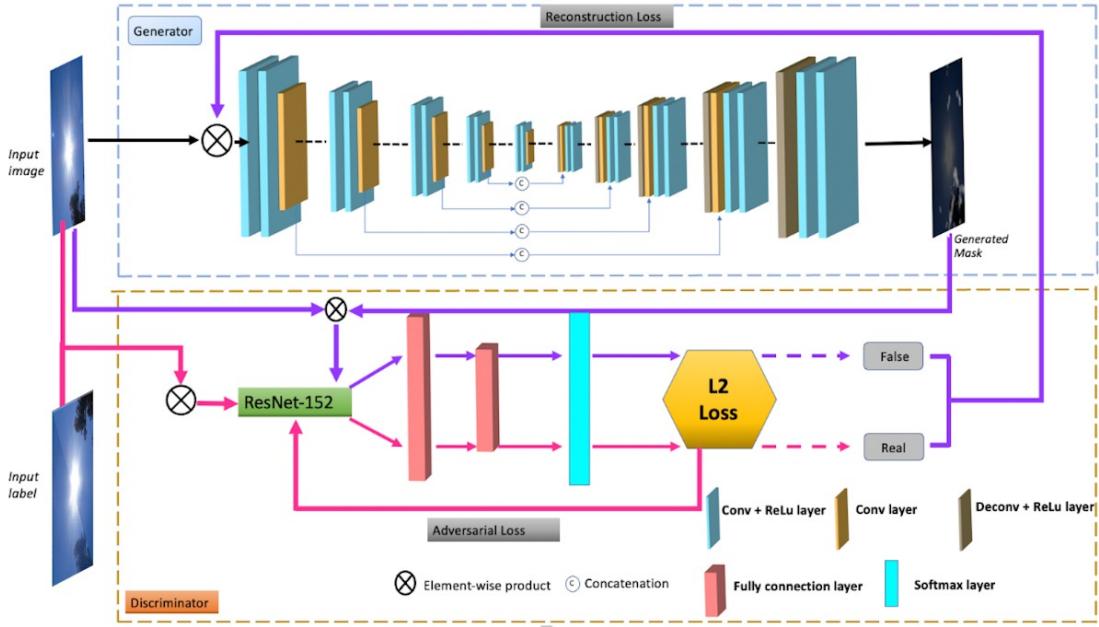


Fig. 3.14: The cGAN architecture used in our model

3.4.1 U-Net Generator

U-net consists of two parts, the first part, which is on the left, is the contracting party, which consists of convolution layers, and then secondly, the expansion part, which up-samples the output of intermediate layers using a combination of convolution and transpose convolutions to recover back the final image. U-net is used for various applications, from segmentation to super-resolution. For the flare removal task, a particular variant of U-net called Contextualized Attentive U-net was chosen. The U-net is based on two principles

- **Squeeze and Excitation module:** The main principle behind this block is first a squeeze Action is applied, I.e., with the help of convolution filters and max pool. The image size is reduced, and this is concatenated to the expanded side of the network after applying a suitable activation function. This is done to calibrate the channel attention of feature maps.
- **Dilated convolutions:**

The main aim of dilated convolutions is to capture a larger field than traditional convolutions, so the kernel's receptive field is increased. In this model, dilations in the bottleneck layer are used. The concept of parallelized dilation It has two branches: the first is a direct connection between the encoder part of the u-net and the decoder part, and the next one consists of a set of parallel dilation blocks with different dilation rates.

After the weights were updated, the generated image is received as shown in Fig3.15. Batch normalisation was also used because it allows for faster training, more stable gradients, and

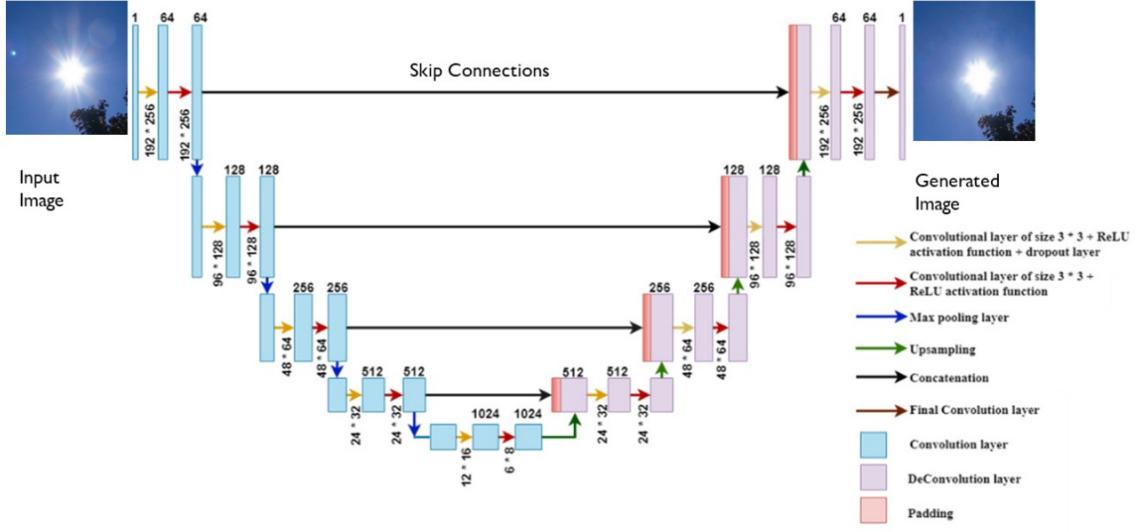


Fig. 3.15: A U-Net architecture with batch normalization [25]

better optimization. It also improves the accuracy of our model, resulting in a better result.

3.4.2 PatchGAN Discriminator

A unique type of discriminator was implemented; rather than discriminating the whole image and applying binary classification, local image patches of smaller sizes are taken and classified each patch into 0 or 1. The probabilities are calculated and averaged to get more accurate results and losses. Since PatchGAN has fewer number parameters, the L2 loss helps us calculate the loss better and trains the model faster.

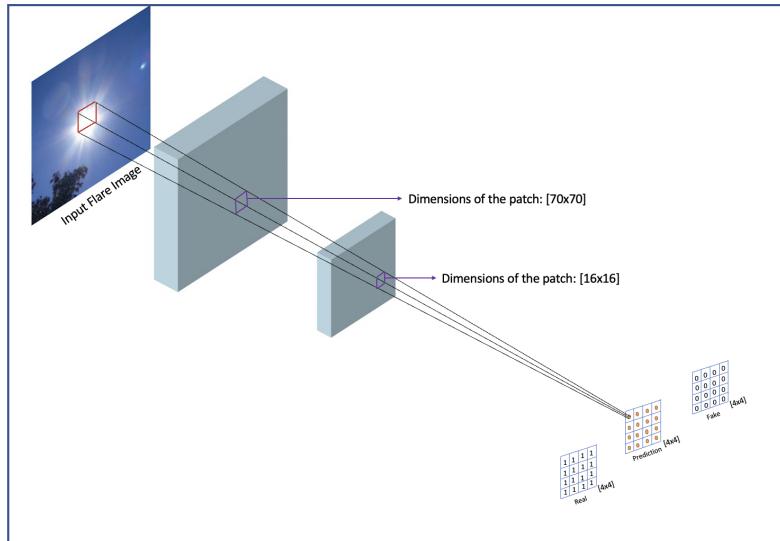


Fig. 3.16: Proposed PatchGAN model detects between generated and ground truth images.

The patches are sent through a convnet model and an FC layer to classify the images. The losses are stored in a list and are then averaged.

3.4.3 Losses

The losses play a pivotal role in tuning the generator to provide better outputs and the discriminator to detect the fakes. A critical loss to give better results is the reconstruction loss. The reconstruction loss is fed into the objective function of the generator, which contains the features learned by the discriminator that help the generator. In our network, a reconstruction loss using ResNet features is used to calculate the pixel distance between ground truth and generator output. It was noticed that it is necessary to ensure that the L2 loss as well as the ResNet feature extraction loss play a pivotal role in updating the parameters of the Generator and Discriminator

$$L_{total}(\varphi, \theta) = \alpha \sum_{(x,Y) \in T} \gamma(\text{ResNet}(x), \text{ResNet}(G_\theta(Y))) + \beta [E_x[\log D_\varphi(x)] + E_Y[\log(1 - D_\varphi(G_\theta(Y)))] + (1 - \alpha - \beta) \sum_{(x,Y) \in T} \gamma(x, G_\theta(Y)) \quad (3.2)$$

From the above equation [26] it is evident that the total loss is a combination of multiple terms. $\alpha \sum_{(x,Y) \in T} \gamma(\text{ResNet}(x), \text{ResNet}(G_\theta(Y)))$ represents the Charbonnier loss which is similar to L2 loss however we add a noise term ϵ which is close to 0.001 , this loss represents the pixel distance between the extracted features of the ground truth image and the generated image. The hyper-parameter term α is a value between 0-1 and was determined experimentally using visual indicators. β is another hyper-parameter experimentally determined. We choose an α value of 0.8 and a β value of 0.2. The other loss term are essential to update the Discriminator and Generator and are standard across most GAN architectures.

The adversarial loss is a binary classifier differentiating between ground truth data and generated data predicted by the generative network. We continuously use this loss to feed it into the discriminator's objective function, and then the losses are finally averaged. This loss is calculated using PatchGAN wherein overlapping patches of size 70x70 are fed into convolutional net and at the output a 16x16 receptive field is received where each element in the matrix corresponds to the probability of the flare being present in the input or not. This probability is then used in a Binary Cross Entropy loss and the output is an essential component of the reconstruction loss.

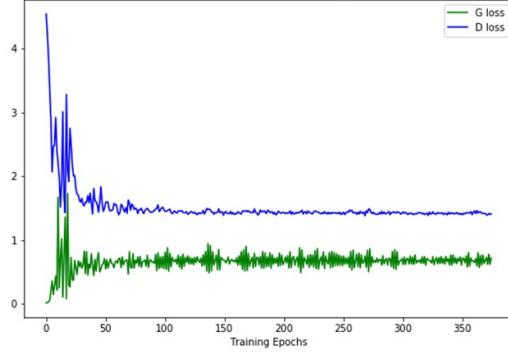


Fig. 3.17: Generator and Descriptor Losses of our model

3.4.4 Metrics

Two metrics that are quite widely used in I2I tasks have been considered. The metrics are Peak Signal-to-Noise ratio (PSNR) and Structural Similarity Index (SSIM).

PSNR:

The peak signal-to-noise ratio is the proportion of an image's maximum conceivable power to the power of corrupting noise that degrades its quality of representation. To calculate a picture's PSNR, it's necessary to compare it to an ideal clean image with the highest potential power. The maximum PSNR for an 8-bit view is 54. It was used to determine the difference between the model's output and the ground truth.

$$MSE = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N [I(i, j) - I'(i, j)]^2 \quad (3.3)$$

$$PSNR = 10 \times \log \left(\frac{255^2}{MSE} \right) \quad (3.4)$$

M = number of rows in image

N = number of columns in image

$I(i, j)$ = pixel value from input image

$I'(i, j)$ = pixel value from generated image

SSIM:

The Structural Similarity Index is a perceptual metric that assesses image quality degradation due to processing such as data compression or transmission losses. It's a complete reference metric that requires two images from the same image capture: a reference image and a processed image. It's well known for video, but it may also be used for still photography. SSIM, in contrast to PSNR, is based on observable picture structures. PSNR is regarded to be less reliable than SSIM. The value ranges between 0 and 1.

$$SSIM = l(I, I')c(I, I')s(I, I') \quad (3.5)$$

- $l(I, I')$ = luminance comparison function
 $c(I, I')$ = contrast comparision function
 $s(I, I')$ = structure comparision function

$$l(I, I') = \frac{(2\mu_I\mu'_I + C_1)}{(\mu_I^2 + \mu'^2_I + C_1)} \quad (3.6)$$

$$c(I, I') = \frac{(2\sigma_I\sigma'_I + C_2)}{(\sigma_I^2 + \sigma'^2_I + C_2)} \quad (3.7)$$

$$s(I, I') = \frac{(\sigma_{II}' + C_3)}{(\sigma_I\sigma'_I + C_3)} \quad (3.8)$$

These metrics are with a ground truth/reference, and however, in the real-world application, a corresponding ground truth making reference metrics a priority to gauge the image quality is absent. There are no established metrics to measure our image for flare applications, which are currently under research.

3.5 Training Setup

3.5.1 Feature Extractor Training

Table 3.1: Using a metrics system, all the previous research results with the method used to achieve it.

Model	Epoch	Optimizer	Loss	Learning Rate	Momentum	Learning Rate, Scheduler
ResNet18	50	Adam	BCE	0.0001	0.95	10,0.1
AlexNet	50	Adam	BCE	0.0001	0.95	10,0.1
MobileNetv2	50	Adam	BCE	0.0001	0.95	10,0.1
VGGNet	50	Adam	BCE	0.0001	0.95	10,0.1
SqueezeNet	50	Adam	BCE	0.0001	0.95	10,0.1

3.5.2 Flare Removal Training

A UNET generator, a Patch-GAN discriminator (similar to pix2pix), and a ResNet18 feature extractor was used for computing the pixel distance used in the reconstruction loss.

Table 3.2: Parametric Data of Flare Removal using two Generator architectures.

Method	Batch Size	Epochs	Generator L.R	Discriminator L.R	Optimizer	Adv.	Recon.
UNet	4	70	1e-5	1e-8	Adam	BCE	L1 loss
VAE	8	70	2e-4	N/A	Adam	BCE	MSE

3.6 Comparison of state-of-the-art models

Identifying the best models for dehazing/denoising/flare removal of an image, a comprehensive analysis of these papers is available in the literature review section. A wide range of methods was studied, starting with conventional methods such as Dark channels before complex deep learning methods like Feature fusion attention networks. To understand each of them, all of the models were implemented in Pytorch on our dataset. Firstly calculation of PSNR and SSIM for the models was done from a validation set of 2000 images. By implementing the SOTA models above, valuable insights into each model were gained and answered the most critical questions. The models designed also incorporated features such as dilations and attention layers and were trained to improve performance. Most of the SOTA architectures are in image dehazing, denoising, and deraining. This makes comparison on flare datasets quite complex, and these models perform relatively poorly

CHAPTER 4

Contribution of Each Student

Aryan James Philip

Contributions

- **Existing Solutions Review:** Studied and analyzed existing solutions in flare removal, denoising, dehazing, and deraining.
- **Model Building:** Built the model for the U-net, discriminator and tested various combinations of Pix2Pix.
- **Model Analysis and Comparison:** Set up baselines for comparing models and documented the code and setup.
- **Dataset Analysis:** Did a deep dive into dataset and analyzed various parameters.

Analysis

To tackle the problem of removing flares from images, previously proposed architectures needed to be examined. For a base, some information was taken from previously devised models and improvements were made on them to ensure better accuracy of the model proposed in this paper. To ensure better working of the model a custom dataset was to be drawn up, which required a lot of research. Multiple improvements later, a model was formulated to ensure that the flare removal architecture proposed in this paper has a better accuracy than previously suggested models.

Amulya Prashant Karkal

Contributions

- **Existing Solutions Review:** Studied previous published reports on flare removal, dehazing, denoising and and deraining.
- **Testing:** Ran multiple flare removal architechctures on the synthetic dataset that was made.
- **Comparison:** Ran multiple dehazing, denoising and deraining architures on a reduced dataset to form comparisons with the dataset created.

Analysis

To make sure that the proposed model that was generated using the generator and the discriminator, is working efficiently, comparing it with a preexisting model is a necessity. Not a lot of research has been done when it come to removal of flares but it is vital to know that the proposed model has greater accuracy. To do so, the previous models were run on the dataset created for this application, to draw a conclusion. The number of epochs, number of paired images and batch sizes were kept constant.

CHAPTER 5

Results and Discussion

5.1 Dataset

Our dataset had images with various dimensions, and after resizing it to a size 256x256, a dip in image quality was observed. This dip resulted in images with different results in terms of PSNR and SSIM. To rectify this, varying image augmentations using the albumenations library was used. However, these steps don't result in much improvement. Also, since our dataset had been collected from open-source resources, many of the images had other visible artifacts like bright spots, text, color shifts, etc. To solve these issues, a lot of image pre-processing was done. Our final dataset was trained on 11,388 images and was validated on 2000 images. Numerous forms of image analysis was also done in terms of histograms and color channel segmentation, and object detection to detect the flares. Also, since image flares were superimposed on the images, it lost information, leading to low input PSNR values.

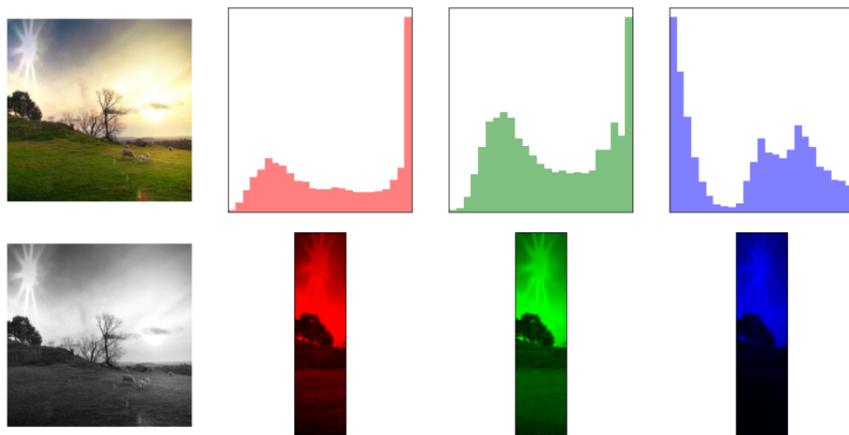


Fig. 5.1: Color channel visualization of an image in the dataset

5.2 Flare Detection

To detect flares, a wide variety of pre-trained CNN networks was used. The observations were quite skewed; however, no overfitting was observed and our test results on most of the networks were entirely satisfactory. The main challenge came in different flares, resulting in a considerable class imbalance. However, it was observed that the ResNet18 model generalized exceptionally well, even on unseen data. A binary classification technique was used to simplify our network to detect flared images from the ground-truth ones. The observations and the training parameters will be mentioned in the table below.

From the above tables, it is clear that while most transfer learning models perform pretty

Table 5.1: Result analysis of Flare classification using SOTA deep learning approaches

Methods	Precision	Recall	F1 Score	Accuracy
AlexNet	91.8	91.5	91.3	91.65
Mobile Netv2	97.6	96.9	97.2	97.3
ResNet18	98.4	95.36	96.64	96.82
AlexNet	94.3	92.5	93.3	93.3
AlexNet	97.8	94.8	96.9	96.2

admirably, the ResNet18 and MobileNetv2 perform the best. While MobileNetv2 performs slightly better across all metrics, the model is quite complex, making it challenging to implement in big datasets. After manually inspecting the results, ResNet18 was finalized. Through ResNet 18, we could implement it in the reconstruction loss, which improves the residual features learned by the model. The ResNet18 model was modified by adding a three-channel convolutional layer before training to visualize the extracted features. Learning strategies like step schedulers, cosine-annealing, and mixed-precision training was used to improve the model. The overall consensus regarding the flexibility of a feature-extractor depends on how well it performs on difficult-to-classify data. Through the many tests, it was concluded that ResNet18 performs best overall.

5.3 Flare Removal Algorithm

Applying the flare removal algorithm to our dataset, it was understood that most I2I techniques do not give the best results. Since our synthetic dataset, our model didn't generalize well on natural images. Often, the issues lay with certain color temperatures as well as there was a lot of change in the HSV space. However, the results on synthetic images were quite admirable, often performing better than most SOTA algorithms and generalizing quite well for night-time photos. Since a U-Net architecture with a feature extractor was used, our model could accurately detect the flares, which helped remove them.

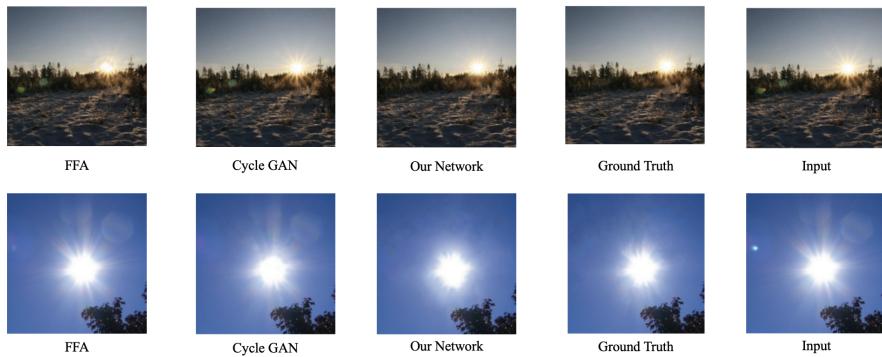


Fig. 5.2: Results of the Flare Removal Algorithm

An evident change in color and contrast is seen from the above results. These results are consistent with most techniques, and post-processing steps are needed to ensure good results. Though our PSNR and SSIM values were quite good, some images gave us a psnr value greater than 30 and a ssim value above 90. However, due to the presence of a lot of outliers, our average psnr and ssim values are lower. Although, our model performed relatively well compared to existing dehazing, deraining, and denoising algorithms. A thorough analysis needs to be conducted on a wide variety of test images to generate accurate conclusions on the effectiveness of our algorithms on a broader scope of image scenes.

Table 5.2: PSNR and SSIM result analysis on two different test datasets

Datasets	PSNR	SSIM	No. of Images
Places 365	27.8	0.873	250
Custom	28.4	0.869	400

5.4 Comparison of Flare Removal Algorithms

Many images were tested on a variety of SOTA dehazing, denoising, and deraining algorithms. Most SOTA algorithms perform relatively poorly on our dataset, with issues like streaking, color shifts, and bright spots appearing in specific images on the test dataset. Since our sample size was enormous, some decent results were achieved with the FFA algorithms and the cycle GAN algorithm.

5.4.1 Feature Fusion Attention Network

Feature fusion attention network is one of the best nets in image dehazing. This report has already looked at the network in detail in the literature review section. This section discusses the network from an implementation standpoint. The network architecture has four key features: pixel attention, channel attention, blocks, and residual connection of groups. When implementing the network on the TensorFlow framework, a local minimum was attained; to get around this model using random weights as direct transfer of weight tensors from Pytorch had proved to be unsuccessful, as some of the layers in Pytorch could not implement in TensorFlow. The other solution to implement the model is converting the model from Py torch to TensorFlow using ONNX were tried. Accomplishing this task, the model converted could only be used for inference, not for training.

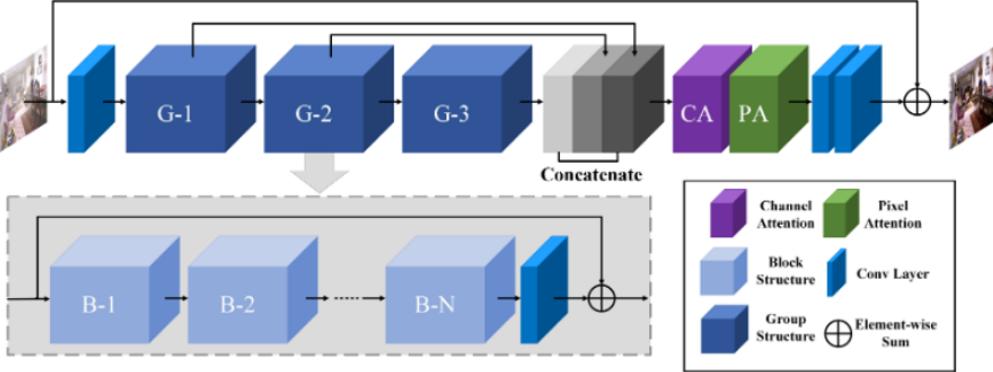


Fig. 5.3: Architecture of FFAnet [11]

This was tested by changing the hyperparameters of the network along with residual blocks and the number of groups. A PSNR of 18.9 and an SSIM of 78.2 was achieved. Although the model performed exceptionally well on specific synthetic images, it performed poorly on natural images. It resulted in significant color shifts on natural images, and in some instances, there was some mild streaking.



Fig. 5.4: Results of FFAnet architecture

The results show that the FFA algorithm did not produce noticeable improvements but rather worsened the photos. The images are put through a cycle GAN architecture to improve the FFA net.

5.4.2 Cycle Generative Adversarial Networks

Cycle GAN is a novel technique of I2I wherein the source and target images are unpaired, and the architecture has two generators and two discriminators. In most ways, it is similar to Conditional Adversarial Networks; however, it differs in a few key areas. The main difference is it has a new loss introduced called Cycle-Consistent Loss. For example, Let's say translation of a sentence from French to English and then back to french has to be done. It is then ensured

that the original sentence is similar to the translated sentence. The difference between both sentences gives us an idea of the Cycle-Consistent Loss. Our implementation was slightly different, wherein some hyper-parameter tuning is used to find the best learning rate.

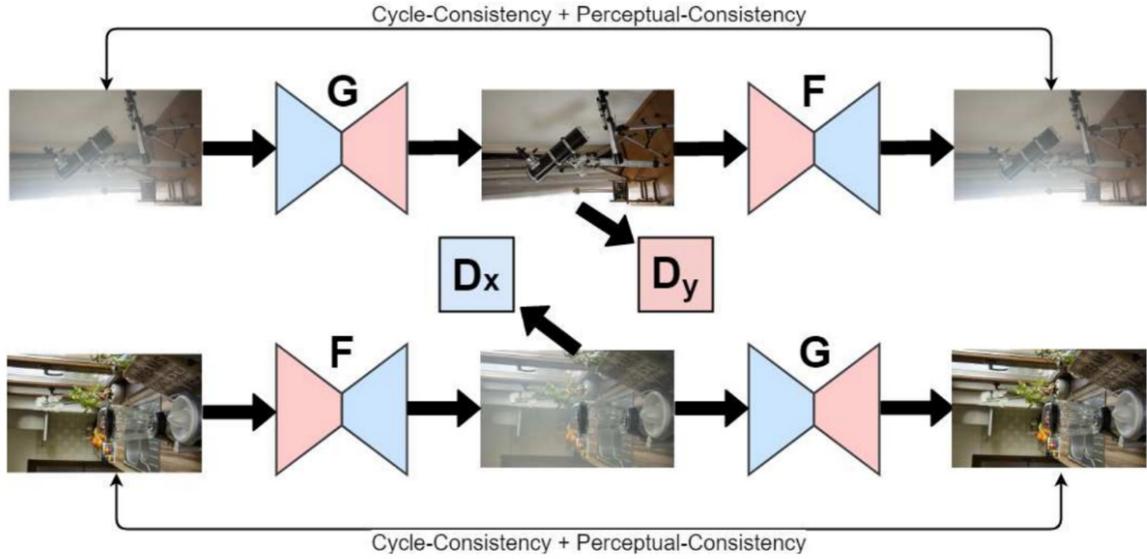


Fig. 5.5: CycleGAN architecture

The idea is that the discriminator of domain B will provide a training output for the generator to generate images with the style of domain B. At the same time, the cycle consistency (reconstruction) loss will nudge the generator to maintain some aspects of the original image (like the shape and structure) in the final output image. Some dimming zones were observed in natural images. However, the model performed well in all conditions when using synthetic images. The issues lie with the amount of training data, the training time, and mode-collapse. Buffers were also added during training for the generator to use references to detect if there is a chance of mode collapse.



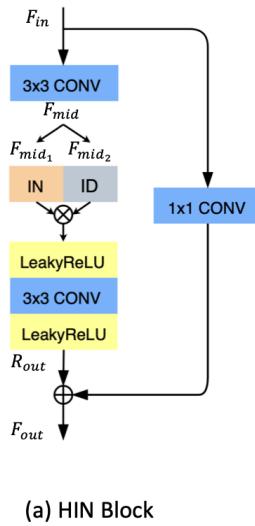
Fig. 5.6: Results of CycleGANet Architecture

CycleGAN operates well, however, it was discovered throughout our testing that the results

on the testing data were heavily skewed, with numerous outliers. As a result, the PSNR and SSIM readings were drastically different.

5.4.3 Half Instance Normalization Network for Image Restoration

Half Instance Normalization Network for Image Restoration (HINet) uses a unique technique of Half-Instance Normalization in which the features are extracted and divided in half to normalize them separately are concatenated together using Cross-Stage Feature Fusion and Supervised Attention Module blocks.



(a) HIN Block

	: Instance Normalization
	: identity

Fig. 5.7: HINET architecture [27]

This technique helps networks to learn only relevant features and discard the rest, which helps in preventing in painting. During our testing it was noticed that there were certain improvements in color and contrast; However, the task of flare removal was only successful in images with less information. A PSNR of 18.2 and an SSIM of 78.2 was achieved.



(a) Flared input image

(b) Generated image

Fig. 5.8: Results of HINET architecture

5.4.4 Nonlinear Activation Free Network

The Nonlinear Activation Free Network (NAF) net proposes a novel technique wherein no activation function is required and existing techniques can be simplified to reduce model complexity and training time.

NAF net also uses multiple U-Net's with inter as well as intra-connectivity. It enhances the image quality and does not result in reduction of color temperature.

During our testing, it was noticed that NAF net performs quite poorly on our dataset, often giving noisy results. However, in certain cases it performs admirably like seen in Fig(5.9). A PSNR of 18 and as SSIM of 78.4 was achieved.

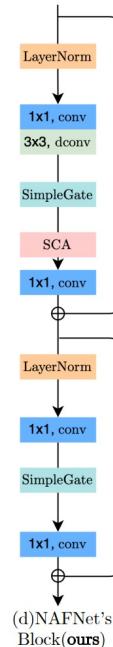


Fig. 5.9: NAFnet architecture [28]



(a) Flared input image

(b) Generated image

Fig. 5.10: Results of NAFnet architecture

5.4.5 Generic Model-Agnostic Convolutional Neural Network

The Generic Model-Agnostic Convolutional Neural Network (GMAN) doesn't use an atmospheric scattering model rather it only depends on the trainable parameters of the model enabling it to learn complex features of flares better. GMAN net has numerous other advantages regarding optimization, input size versatility, efficiency, and complexity of the network.

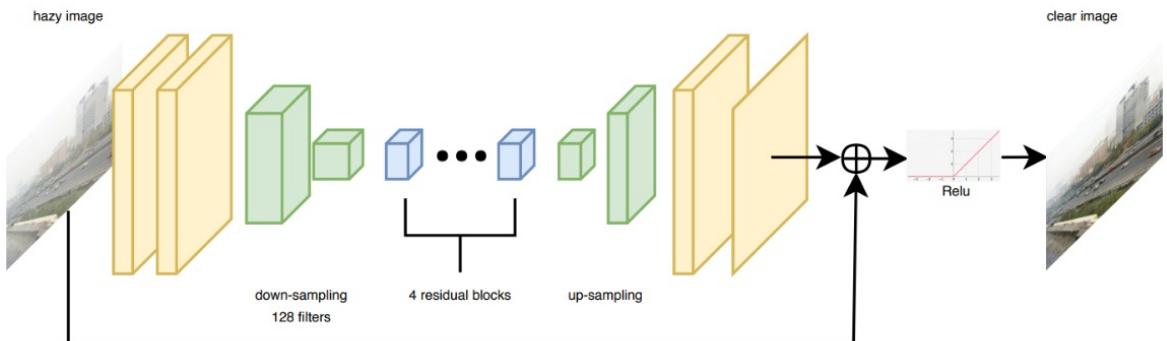


Fig. 5.11: GMAN architecture [10]

However, from Fig (5.11) seen below GMAN net was developed for the purpose of dehazing and falls short when trained on flared images. The GMAN net was tested on 50 images and achieved a PSNR of 18 and an SSIM of 80.



(a) Flared input image

(b) Generated image

Fig. 5.12: Results of GMAN architecture

Various models ranging from dehazing, deraining, and deblurring was tested and it was observed that all these algorithms performed quite poorly with no improvement. In some cases, the images became worse. The metrics of all the algorithms will be compared below.

Table 5.3: Quantitative analysis of Flare Removal using SOTA deep learning approaches

Methods	Test Images	PSNR	SSIM
CycleGAN	50	21.6	0.812
FFA net	50	18.9	0.796
HI-Net	50	18.2	0.782
NAF Net	50	18	0.784
GMAN Net	50	18	0.80

5.5 Result Analysis and Discussion

To further understand the results, it is necessary to dissect each model's results as well as their failure cases. From the above quantitative and visual results, some conclusions can be drawn.

5.5.1 Qualitative Analysis

The recent work in the broader space of image restoration tasks do not attempt to tackle the general flare removal problem. They handle only specific types of flares like glare spots, and they build specific loss functions and heuristics to remove them. These models aren't able to remove a variety of flare types like streaks and halos and the resultant PSNR and SSIM values are very similar to the input image. Furthermore, even image dehazing and deraining models which have been deeply analyzed, do not work on flares well, often blurring the image and in some cases a significant shift in color and contrast. Since, the work proposed by us was trained across multiple scenes and different types of flares, it is able to robustly remove lens

flare of various shapes, colors, and locations on diverse synthetic images. It even generalizes to multiple light sources.

Cycle GAN performs better than most other dehazing and deraining algorithms because it learned diverse flare artifacts in the training process. However, in some cases it fails to remove large flare spots and cannot detect multiple light sources. Since, our work proposes a feature extractor in the loss function it is able to generalize across different flare positions and intensities. Our flare removal algorithm is also able to remove flares from images without a visible light source. Images without flares were also used as input to our model to make it more robust against barely visible flares.

FFA net also performs well in cases where there is less background information and detail. In most cases, it changes the color drastically and induces some blurring. Though, it does perform significantly better than other more complex dehazing models.

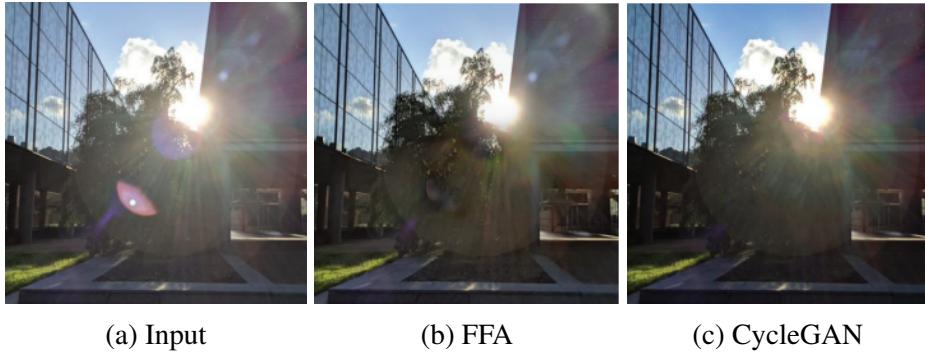


Fig. 5.13: Result Analysis

5.5.2 Quantitative Analysis

From table 5.2 and 5.3, it can see that there is a significant dip in PSNR and SSIM when compared to our model. This shows that even on larger test sets our model outperforms existing dehazing, denoising, and deraining algorithms.

5.5.3 Failure Cases

Since flare removal is a new topic, there is still a lot of research to be done to improve pre-existing work. It was noticed that, our model fails in certain scenarios. A common observation, was when the flares were too bright our results were either extremely blurry or there was no change in the output. Also, in cases where there were multiple flares of different shapes and sizes our model didn't remove the flares successfully. It also failed when the source of the flares weren't evident. Some of the failure cases will be shown below.



Fig. 5.14: Failure case when the flare and light source are disjointed



Fig. 5.15: Failure case when the light source is too bright

CHAPTER 6

Conclusion and Future Scope

6.1 Conclusion

Through this study, a thorough comparative analysis of the SOTA models is presented. This work considered traditional metrics such as PSNR and SSIM on a completely new dataset generated synthetically. Our work is novel as addition to previous work has been done by improving the loss function and doing a deep analysis on the feature extraction process. There is still a long way to go in terms of model training, evaluation, image sizes and building an end to end pipeline for flare removal.

6.2 Future Scope

Though a lot of time and effort was put into this project, some areas require additional work and could be taken up in the future. One of the main tasks is deploying the model and making the model adaptable to different input image sizes, thereby enabling the project to be integrated with applications such as self-driving car software. It would also expose the shortcomings of the model, thereby enabling the betterment of the model.

Furthermore, work can be done to develop flare related metrics which would give us a baseline upon which future work can be done. Setting up a standard dataset of a mixture of high quality natural and synthetic images, will enable continuous research and development as seen in the field of image dehazing. Well documented material, code, and videos would go a long way to improve the quality of research.

REFERENCES

- [1] N. Solis. “Hazy skies, poor air quality: Is port congestion worsening l.a. pollution.” (2021), [Online]. Available: <https://www.latimes.com/california/story/2021-11-11/is-port-shipping-congestion-worsening-la-pollution>. (accessed: 13.06.2022).
- [2] M. Myllykoski, “On gpu-accelerated fast direct solvers and their applications in image denoising,” Ph.D. dissertation, Aug. 2015.
- [3] K. S. BA. “Possibility of low pressure; thunderstorms with rain in state for five days, precautions to take.” (2022), [Online]. Available: <https://keralakaumudi.com/en/news/news.php?id=786389&u=possibility-of-low-pressure-thundershowers-with-rain-in-state-for-five-days-precautions-to-take-786389>. (accessed: 13.06.2022).
- [4] V. Thakur. “What is lens flare.” (2022), [Online]. Available: <https://www.scienceabc.com/innovation/what-is-lens-flare.html>. (accessed: 13.06.2022).
- [5] A. Criminisi, P. Perez, and K. Toyama, “Region filling and object removal by exemplar-based image inpainting,” *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1200–1212, 2004. DOI: 10.1109/TIP.2004.833105.
- [6] S. Lee and E. Eisemann, “Practical real-time lens-flare rendering,” in *Proceedings of the Eurographics Symposium on Rendering*, ser. EGSR ’13, Zaragoza, Spain: Eurographics Association, 2013, pp. 1–6. DOI: 10.1111/cgf.12145. [Online]. Available: <https://doi.org/10.1111/cgf.12145>.
- [7] P. Vitoria and C. Ballester, “Automatic flare spot artifact detection and removal in photographs,” *Journal of Mathematical Imaging and Vision*, vol. 61, no. 4, pp. 515–533, 2019.
- [8] X. Qiao, G. P. Hancke, and R. W. Lau, “Light source guided single-image flare removal from unpaired data,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 4177–4185.
- [9] Y. Wu, Q. He, T. Xue, *et al.*, “How to train neural networks for flare removal,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2239–2247.
- [10] Z. Liu, B. Xiao, M. Alrabeiah, K. Wang, and J. Chen, “Generic model-agnostic convolutional neural network for single image dehazing,” *arXiv preprint arXiv:1810.02862*, 2018.
- [11] X. Qin, Z. Wang, Y. Bai, X. Xie, and H. Jia, “Ffa-net: Feature fusion attention network for single image dehazing,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 11908–11915.

- [12] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.
- [13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [14] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.
- [15] F. Chabert, “Automated lens flare removal,” 2015.
- [16] M. A. Bagheri. “Conditional generative adversarial nets.” (2019), [Online]. Available: <https://medium.com/@ma.bagheri/a-tutorial-on-conditional-generative-adversarial-nets-keras-implementation-694dcafa6282>. (accessed: 20.05.2022).
- [17] S.-H. Psang. “U-net review: U-net(biomedical image segmentation).” (2018), [Online]. Available: <https://towardsdatascience.com/review-u-net-biomedical-image-segmentation-d02bf06ca760>. (accessed: 20.05.2022).
- [18] T. Ganokratanaa, S. Aramvith, and N. Sebe, “Unsupervised anomaly detection and localization based on deep spatiotemporal translation network,” *IEEE Access*, vol. 8, pp. 50 312–50 329, 2020. DOI: 10.1109/ACCESS.2020.2979869.
- [19] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.
- [20] R. S. Ghorakavi, “Tbnet: Pulmonary tuberculosis diagnosing system using deep neural networks,” *arXiv preprint arXiv:1902.08897*, 2019.
- [21] A. NG. “Image net classification with deep convolutional neural networks.” (2012), [Online]. Available: <https://coursera.org/share/1fe2c4b8b8d1e3039ca6ae359b8edb3>. (accessed: 20.05.2022).
- [22] W. Almeida, F. Andaló, R. Padilha, *et al.*, “Detecting face presentation attacks in mobile devices with a patch-based cnn and a sensor-aware loss function,” *PLoS ONE*, vol. 15, e0238058, Sep. 2020. DOI: 10.1371/journal.pone.0238058.
- [23] Pawan. “Vgg-16—cnn model.” (2020), [Online]. Available: <https://www.geeksforgeeks.org/vgg-16-cnn-model/>. (accessed: 20.05.2022).
- [24] A. Pujara. “Classification with mobilenet.” (2020), [Online]. Available: <https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fb2cd470>. (accessed: 20.05.2022).
- [25] N. Ibtehaz and M. S. Rahman, “Multiresunet: Rethinking the u-net architecture for multimodal biomedical image segmentation,” *Neural Networks*, vol. 121, pp. 74–87, 2020.

- [26] A. Lucas, S. Lopez-Tapia, R. Molina, and A. K. Katsaggelos, “Generative adversarial networks and perceptual losses for video super-resolution,” *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3312–3327, 2019.
- [27] L. Chen, X. Lu, J. Zhang, X. Chu, and C. Chen, “Hinet: Half instance normalization network for image restoration,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 182–192.
- [28] L. Chen, X. Chu, X. Zhang, and J. Sun, “Simple baselines for image restoration,” *arXiv preprint arXiv:2204.04676*, 2022.

ANNEXURE 1

PO & PSO Mapping

Student Name:

AMULYA PRASHANT KARKAL
ARYAN JAMES PHILIP

Registration No.:

180929142
180929144

PO	✓ Tick	Page. No.	Section No.	Guide's Observation
PO1	✓	9-12, 21-23	3.1,3.4.3	Applied engineering fundamentals to model building
PO2	✓	4-8,	2.1-2.2	Conducted literature review and existing solutions
PO3	✓	17-23	3.4.1-3.4.6	Designed novel flare removal model
PO4	✓	26-36	5.4.1-5.5.3	Conducted comparative analysis on a reduced dataset
PO5	✓	22,23	3.5.1-3.5.2	Used PyTorch, tensorflow
PO6				
PO7				
PO8				
PO9	✓	24-25	4	Worked in a team with individual goals and contributions different domains during the project
PO10	✓	4,26,27	2.3, 5.2, 5.3	Detailed report and presentation on the project
PO11				
PO12	✓	26-36	5.1-5.5	Used online resources to learn new skills during the project

PSO	✓ Tick	Page. No.	Section No.	Guide's Observation
PSO1				
PSO2				
PSO3	✓	26-35	5.1-5.5	Used PyTorch for model building analysis and data sharing

Signature of Student:

Name and Signature of Guide:

Date:

ANNEXURE 2

PLO Mapping

Student Name:

AMULYA PRASHANT KARKAL
ARYAN JAMES PHILIP

Registration No.:

180929142
180929144

PLO	✓ Tick	Pg. No.	Section No.	Guide's Observation
C1	✓	9-12, 21-23	3.1,3.4.3	Applied engineering fundamentals to model building
C2	✓	26	5.1-5.5	Designed a model using first principles of mathematics and deep learning
C3	✓	27-28	5.3	Used Deep Learning Techniques to solve the problem
C4	✓	4-7	2.1-2.2	Did thorough analysis of existing techniques
C5				
C6				
C7				
C8				
C9				
C10				
C11				
C12	✓	19-23	3.4-3.6	Used robotics lab to aid in the project
C13	✓	19-23	3.4-3.6	Used GPU's to maid and building the model
C14				
C15				
C16				
C17	✓	26-36	5.1-5.5	Detailed report and presentation of the project
C18	✓	26-36	5.1-5.6	Used online resources to learn new skills during the project

Signature of Student:

Name and Signature of Guide:

Date:

ANNEXURE 3

IET Outcomes

1. Explain the steps you considered to investigate and define the problem in your project work (C4, evaluate level)

To define the problem statement and objectives to be achieved for the project, we conducted a review of the existing published literature to understand the common points of classification among the proposed solutions, and graded the various solutions in the respective categories. We also conducted market research by visiting various farms to understand their problems and conducted a case study on one of the most currently popular solutions that we would then set as the benchmark to surpass in our current project.

2. Discuss the science, mathematics, statistics, engineering principles and other basic technology you identified for design (Mechanical, Electronic, Physics, Chemistry, Automation) in your project work. (C1, C2, C3, Application, Analysis, Evaluation of Science and Mathematics in the project)

Artificial intelligence, statistics and computer vision were the main fields of study and research in the project. Other components were the occurrence of flares and different ways to generate the dataset.

3. Have you considered the Environmental and Sustainability limitations in your project work? (C7, evaluate)

No we haven't considered the environmental implications since flares do not generate harmful gases. However, we considered the emissions of CO₂ when we were training our models.

4. Have you considered ethical, health, safety, security, and risk issues; intellectual property; codes of practice and standards while addressing these issues in your project work? If so, Explain in detail. (C5, create)

We have used the Do not Repeat yourself approach (DRY) while programming and made our code modular, also effort has been taken to contribute to open source community. We also have given credit when we have reused code.

5. What were the aesthetic issues faced and how it was addressed in your project in the design phase?(C5, analysis)

No aesthetics were addressed as there weren't any in our field of research.

6. Were there any health issues considered during design process? How it was addressed in your project in the design phase?(C5, create)

There weren't any health implications during the design phase since our project focuses on Neural Network development.

7. What were the safety, security and risk issues considered in the design stage?(C10, create)

Since flares could play a role in real time applications such as self-driving , we considered the implications of our research for such applications.

8. Have you come across intellectual property issues in the project phase?(C5, create)

There weren't any intellectual property issues faced in the project.

9. What are the codes of conduct and standards you needed to use in design phase and in other phases of your project as well? (It may include codes of practice and standards for safety, security, health, risk) Explain the legal issues, ISO standards, IEC standards, etc.(C8, evaluate)

Since the project was conducted in an online mode, we needed to be punctual with our submission deadlines and reports.

10. What were the professional ethics needed to be followed in general while you are doing the project?(C8, evaluate)

Some of the professional ethics needed to be followed in generally during the project are: accountability, honesty, loyalty and respects for others.

11. Do you think ethics and professionalism needs to be paid attention by students during study? If, yes, explain how it can be inculcated/introduced/implemented?(C8, evaluate)

Ethics and professionalism are an integral part for any work environment.Implementing ethics and professionalism will manifest the respect for fellow members and their work and most importantly respect for each other's time which is of paramount significance. We inculcated this by following a schedule each week, honesty,respecting each other and receiving suggestions and feedback from our mentor.

12. Do you think environmental and sustainability limitations; ethical, health, safety, security, and risk issues; intellectual property; codes of practice and standards are sufficiently covered in the courses you have studied in your curriculum?(C8, evaluate)

Environmental and sustainability limitations; ethical, health, safety, security, and risk issues; intellectual property; codes of practice and standards were taught to us in our first year and was also inculcated into our course to some degree.

13. Have you gone through online classes, or a crash course in which you are familiarized with intellectual property rights as well as risk issues in professional environment?(C8,

evaluate)

Yes, we have gone through coursera courses in which you are familiarized with intellectual property rights as well as risk issues in professional environment.

- 14. In the beginning of your project did you evaluate environmental effects and sustainability factors in your work? (C7, evaluate)**

Our project has no environmental implications.

- 15. Did you address the limitations of your project work and have you improved the results through continuous improvements in your project work? (C5, create)**

Yes, the limitations were addressed and we followed an agile methodology for continuous improvement over time.

- 16. How did you plan your project, deadlines, maintaining dairy of each stage and improved the quality of the project? (C14, understand)**

Implementing ethics and professionalism will manifest the respect for fellow members and their work and most importantly respect for each other's time which is of paramount significance. We inculcated this by following a schedule for each week, honesty, respecting each other and receiving suggestions and feedback from our mentor.

- 17. Are you aware of the ethical clearance when you work in the field of health/medical applications? (C8, evaluate)**

No, I haven't delved into the field of health/medical applications and don't know about the ethical clearance.

- 18. Did you adopt any quantitative technique for any engineering activity related to your project? (C3, evaluate)**

The project's main focus was to setup a comparision scheme and to tune the loss for its improvement. Hence, there wasn't any need of quantitative techniques.

- 19. What were the elements of your project work which addresses sustainable development and were you able to apply quantitative techniques to analyze and achieve your project goals? (C7, evaluate)**

There aren't any elements of our project work which addresses sustainable development and quantitative techniques were not adopted in this project.

- 20. Did your project need the understanding of relevant legal requirements governing engineering activities you carried out as a part of your project work? Explain in detail. (C8,**

evaluate)

No, this project did not have any requirements for knowledge on legal requirements for engineering activities.

21. What are the legal, ethical practices you followed while working on project? (C8, evaluate)

The project was carried out in online mode and punctuality was important.

22. Are you sure that you abide IPR/copy right issues? (C15, apply)

Yes, this project does abide by the IPR/Copyright issues.

23. What online course you attended to improve your communication skills, Report writing, Oral presentation, Software used for writing report? (C17, apply)

There was no online course taken up by us to improve our communication skills, Report writing, Oral presentation, Software used for writing report.

24. In your project, was it needed to tackle risk issues, including health and safety, environmental and commercial risk, and of risk assessment and risk management techniques? Explain in detail. (C5, create)

There were no risk issues, including health safety, environmental and commercial risk, and of risk assessment faced by this project.

25. How is the organization addressing a fire accident/human safety when working with machines? (C9, evaluate)

We are not aware of the fire accident/human safety precautions carried out at the organization.

26. Process of teamwork. How each of you are involved in the team? What part the work is addressed by you? (C16, evaluate)

We split the work for each model and kept up with each other by using GitHub and had regular meetings on Teams.

27. Have you filed patent, IPR, or published your work? Give more details. (C17, evaluate)

We have not filed for a patent, IPR or published this work.

28. How you documented the literature review, your analysis on their results, discussion with the guide and team members, provide the documents on weekly basis. Put as one chapter in final report. (C4, evaluate)

Since the project was being carried out on online mode, we used GitHub to the fullest purpose by making a calendar, the issues faced and needed improvement. GitHub saves all this data in

chronological order and hence it was very helpful.

29. **Have you sensitized about inclusion and diversity in the team? If yes, what are the diversification in the team in terms of religion, gender, ethnicity, etc? (C11, apply)**

Our primary focus on creating an end to end flare removal model, so we did not sensitize about inclusion and diversity in the team.

30. **How were you able to keep yourself updated with the technology? How you incorporated advanced technology in your project. (C18, lifelong learning)**

We were up to date with the current technologies with the help of internet, and we also included some of the latest technology in our project.

31. **Which are the laboratory skills you found applicable to your project? Explain. (C12, apply)**

Robotics labs were applicable to this project.

ANNEXURE 4

Project Work Classification

Student Name:

AMULYA PRASHANT KARKAL
ARYAN JAMES PHILIP

Registration No.:

180929142
180929144

Note: Use a tick mark to specify under which domain your project work falls into.

Table 1: classification based on project domain classification

Domain	✓ Tick
Product	
Application	✓
Review	✓
Research	✓
Management	

Note: Use a tick mark to specify societal impacts you considered during your project work.

Table 2: classification based on societal consideration

Domain	✓ Tick
Ethics	
Safety	
Environmental	
Commercial	✓
Economical	✓
Social	

Signature of Student:

Name and Signature of Guide:

Date:14/06/2022

ANNEXURE 5

Project Details

Student Details			
Student Name	AMULYA PRASHANT KARKAL		
Registration Number	180929142	Section / Roll. No.	B / 28
Email Address	amulya.karkal@learner.manipal.edu	Phone No(M)	7219608522
Student Name	ARYAN JAMES PHILIP		
Registration Number	180929144	Section / Roll. No.	B / 29
Email Address	aryan.philip@learner.manipal.edu	Phone No(M)	9611216111
Project Title	Single Image Lens Flare Removal Using Deep Learning		
Date of Synopsis	18/02/22	Date of Final Presentation	17/06/22
Internal Guide Details			
Name of the Guide	Dr Asha C S		
Full contact address with pin code	Department of Mechatronics, Manipal Institute of Technology, Manipal – 576104 (Karnataka State), India		
Email Address	asha.cs@manipal.edu		

PLAGIARISM CHECK

11%	SIMILARITY INDEX	6%	INTERNET SOURCES	7%	PUBLICATIONS	3%	STUDENT PAPERS
<hr/>							
PRIMARY SOURCES							
1	Submitted to Manipal University Student Paper	2%					
2	Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, Jonathan T. Barron. "Unprocessing Images for Learned Raw Denoising", 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019 Publication	1%					
3	Siyuan Li, Xiaochun Cao, Iago Breno Araujo, Wenqi Ren et al. "Single Image Deraining: A Comprehensive Benchmark Analysis", 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019 Publication	<1%					
4	www.coursehero.com Internet Source	<1%					
5	Submitted to Sikkim Manipal University Student Paper	<1%					
<hr/>							