Supervised Multi-Domain GAN for Low Light Image Synthesis

Project Report

Group 845

Aalborg University

Electronics and IT

AALBORG UNIVERSITY

STUDENT REPORT

Electronics and IT

Aalborg University

http://www.aau.dk

Title:

Supervised Multi-Domain GAN for Low Light Image Synthesis

Theme:

Computer Vision

Project Period:

Spring Semester 2022

Project Group:

Group 845

Participant(s):

Sanket Suresh Kokane

Shristy Shah

Supervisor(s):

Andreas Aakerberg

Kamal Nasrollahi

Copies: 1

Page Numbers: 37

Date of Completion:

May 25,2022

Abstract:

Low light image enhancement deep learning networks need enormous amount of images for training. State of the art low light enhancement models use a mixture of real low light image datasets and synthetically degraded images by adjusting gamma and adding noise. However, most of the existing datasets are insufficient and the degradation pipelines do not capture the true essence of real low light images. RELLISUR was created to overcome this challenge and provides a large scale real low light to real normal light image dataset. This still does not address every use case in the set of low light image enhancement problems. In order to extend RELLISUR and provide a sophisticated image degradation method, we present a supervised multi-domain GAN for normal light to low light translation. Our model takes real normal light images and generate low light images with specified under exposure value ranging from -2.5EV to -4.0EV. Upon extensive experiments, Our model was also found to provide better and variety of degradation, closer to real low light images.

Preface

This project is made by group 845 from Aalborg University on the 8th semester of Vision, Graphics and Interactive Systems. This project consists of this report and a shortcut to groups Github page. The group would like to thank Andreas Aakerberg and Kamal Nasrollahi for supervising the project.

Aalborg University, May 25, 2022

Sanket Suresh Kokane

<skokan21@student.aau.dk>

Shristy Shah

<sshah21@student.aau.dk>

Contents

Acronymlist

CCTV	Closed-Circuit Television
CMOS	Complementary Metal-Oxide-Semiconductor
DISTS	Deep Image Structure and Texture Similarity
EV	Exposure Value
GAN	Generative Adversarial Network.
HD	High Definition
HSV	Hue,Saturation,Value
ISO	International Organization for Standardization
LED	Light Emitting Diode
LOL	Low-Light Dataset
LLNet	Low-Light Net
LPIPS	Learned Perceptual Image Patch Similarity
MBLLEN	Multi-Branch Low-Light Enhancement Network
PSNR	Peak Signal to noise ratio
RELLISUR	REal Low-Light Image SUper-Resolution
SNR	Signal-to-Noise Ratio
SSDA	Stacked Sparse Denoising Auto-encoder
SSIM	Structural Similarity Index measure

Chapter 1

Introduction

There are a lot of scenarios where image capture in a low-light environment is not only unavoidable but paramount. CCTV cameras capture at a much lower quality than what can be used for image processing tasks like face detection. This is done to account for the continuous footage and memory. Thus they suffer the most during nighttime or in low light environments. Sometimes mishaps that can occur at night are monitored by CCTV cameras or dash cams on cars, fail to provide sufficient evidence. Drones or robots that work in low light environments like mines or underwater may not work as efficiently as they do in presence of light. Low light image enhancement has multiple applications in military reconnaissance, medical surgeries, autonomous vehicles, etc where footage captured at night or in an extremely low light environment becomes unusable. Capturing low light images still remains an issue for a variety of use cases but modern image reconstruction methods compliments this by enhancing the low light footage to some degree of normal light to render it usable. This area of research has seen large scale efforts to develop models that are exceptionally accurate. However, most of these methods do not use actual low light images but synthesize a normal light image dataset by degrading it. Many prominant image enhancement network like LLNet creates its synthetic low light images by adding noise and adjusting contrast [18], RetinxNet does so by adjusting the illumination and reflectance of normal light images to match that of low light images [31] and MBLLEN generates low light images by adding Poisson noise and gamma adjustment [19]. Models trained using these synthetic low light images generally do not pan to work well with real low light images. The downscaling may alter natural noise and other natural characteristics of low light images, which in turn results in color distortions, overexposure, excessive noise, etc in the generated normal light images. Aakerberg et al [1] created a dataset with real low light low-resolution

images and corresponding normal light high-resolution images to curb the issues with low light enhancement and super-resolution over synthetic low light images. Although the dataset encompasses both indoor and outdoor scenes, due to obvious constraints, there is a lack of animate objects and discrete geological and environmental conditions. Any enhancement model conforming to a very specific problem, demands appropriate data for the sake of precision. To service these needs and extend datasets similar to RELLISUR, we propose a multi-domain supervised GAN model that can generate low light images of varying exposure values.

1.1 Characteristics of Low Light Images

Natural low light images pose certain characteristics that define the natural features of the subject of the image. Settings in the capturing device define some of these characteristics. The camera sensor is responsible for capturing light photons that make up the image. A sensitive camera sensor can improve the details in the captured image. However this increased sensitivity also impacts cost, so better sensors are typically found on higher-end devices. A sensitive image sensor can be coupled with an advanced image signal processor (ISP) that can deliver high-quality images in challenging conditions. CMOS sensors and ISP modules are optimized specifically for video surveillance in low-light environments by manufacturers. An ISP module is responsible for color balance, white balance, exposure levels, gamma correction, dynamic range, etc.

Another major challenge in low light imaging is having enough light falling onto the camera sensor which becomes difficult with higher resolution sensors. Sensors need the light to fall onto the individual pixels to create a sufficient charge to set values for each picture element. An external source of illumination can make up for this lack of light but in the case of security cameras or military recognizance operations, the presence of an illuminating object defeats the purpose of stealth.

Some settings that enable capturing low light images are ISO and shutter speed. Camera sensitivity to light increase with ISO. Thus when capturing in low light settings, high ISO will fetch more light and better images. Figure 1.2 shows images with increasing ISO values. The speed at which the camera closes its shutter is known as the shutter speed. It controls the duration for which the light can fall onto the sensor. Figure 1.3 shows the effect of various shutter speed settings. However, in both these cases, random

noise is introduced in the resulting images. Higher ISO and lower shutter speed will disrupt the Signal to Noise Ratio (SNR). SNR is the presence of unwanted noise along with signal which is the actual information about the picture. Along with this, a lower shutter speed will also induct motion blur in the images in a dynamic environment. Aperture is the opening through which the light passes on the sensor. Adjusting the aperture can control the amount of light that falls onto the camera sensor. With larger aperture the depth of field is smaller which might be undesirable. An EV step can be best described using a triangle with ISO, Aperture, and Shutter Speed being the three vertices. Figure 1.1 shows the exposure triangle and how the three factors affect the exposure value of the image. The exposure value is calculated as $log_2 \frac{N^2}{t}$ where N is the camera lens f-stop and t is the exposure time in milliseconds.

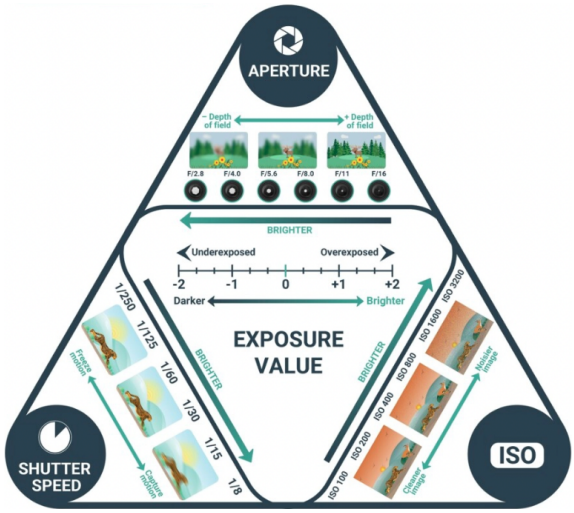


Figure 1.1: The exposure triangle[6]

Figure 1.2: Effect of ISO settings on an image[9]

Figure 1.3: Effect of shutter speed on an image[10]

The presence of noise is another characteristic of low light images. This noise is the result of random altercation of brightness and color and often occurs in low light conditions. Noise can also occur due to heat near the camera sensor. This noise compliments the subject of low-light images. Noise is almost unavoidable in low light imaging and thus must be present in images that will be used to train low light enhancement models for it to be able to enhance real low light images.

Post-processing involves gamma correction and white balancing. Gamma correction is the way to correct the light in images corresponding to human perception. The camera sensor considers two photons on the same pixel as twice the intensity. On the contrary, humans record this as only a fraction brighter. To correct this and not allow the image to white out, gamma correction is an important step in the post-capture optimization stage. Figure 1.4 show images with and without gamma correction. White balancing works the same as adjusting the colors with respect to the light source. The camera and humans perceive colors differently needing white balancing. Numerous color models try to bridge this gap where Hue, Saturation, Value(HSV) comes closest to human perception of color. Figure 1.5 shows an image before and after white balancing. Both of these adjustments are necessary in low light imaging as to not lose the colors and prevent the image from appearing very saturated or completely void of any colors.

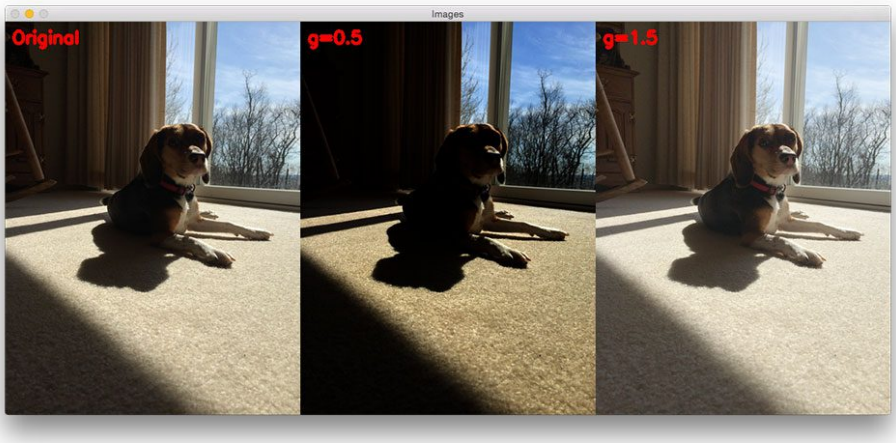


Figure 1.4: Effect of Gamma Correction on an image[24]

Figure 1.5: Before White Balancing(left), After White Balancing(Right)[32]

1.2 Initial Problem Statement

In summary, low light images are currently being generated using either degradation pipelines or by adjusting camera settings. Degradation pipelines try to match the features of the normal light image like histogram or gamma to that of low light images while simultaneously adding random noise. Although this process is effective in its ways, it differs from real low-light images based on a variety of factors. Manually capturing low light images by adjusting ISO, Shutter speed, and Aperture is a very tedious and time-consuming method and also requires considerable post-processing. This process gets even more laborious considering the sheer amount of images that are needed by state-of-the-art deep learning models. Both these approaches have their backlogs with the former being practical but not precise while the latter is precise but impractical. The demand and range of applications of low light images drive the progress of technology for creating this large-scale dataset. RELLISUR provides a large dataset with paired real low light and normal light images but is not enough for every low light image enhancement problem based on different environments, indoor-outdoor settings and due to ethical reasons, human faces and animals. Thus there is a need to extend low light datasets like RELLISUR but in a more practical way than simply capturing more images.

A preliminary problem statement has been set as follows:
How can a normal light image be synthesized to a realistic low light image using a deep learning model?

Chapter 2

Related Works

2.1 Rellisur

RELLISUR [1] is a first of its kind large scale dataset of paired real low light low-resolution images and normal light high resolution with the objective to bridge the gap between low light enhancement and super-resolution problems. The images were captured using a Canon EOS 6D camera equipped with a Canon 70-300mm L IS USM zoom lens. For low light images, the auto bracketing mode was used to capture under-exposed images. To further generalize, two different ranges were used going from -4.5 EV to -2.5 EV and -5.0 EV to -3.0 EV steps, and then averaged out. The dataset contains 12750 images grouped into 850 distinct sequences, with each sequence containing x1, x2, x4 scaled normal light images and five real low light images with exposure levels -5.0 EV to -3.0 EV.

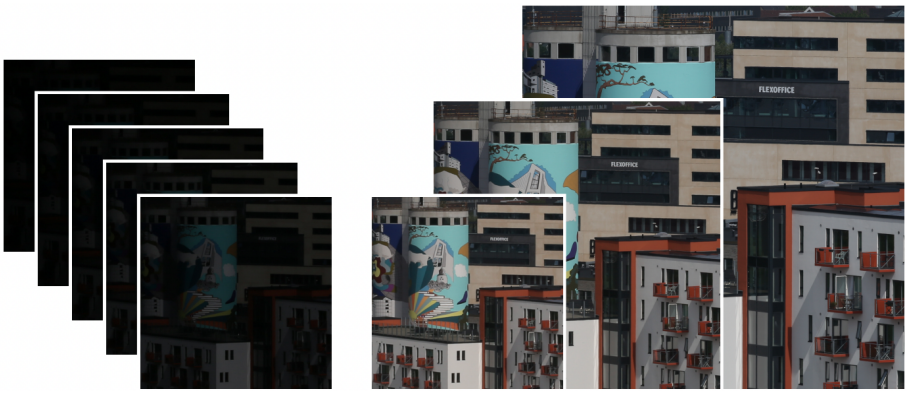


Figure 2.1: Single sequence of images with exposure levels -5.0 to -3.0 (left) and scale levels x1, x2 and x4(right) for the same image[1]

2.2 Low Light Image Synthesis and Enhancement

In this section, we look at low light image enhancement models and their degradation pipelines.

2.2.1 LLNet

Lore et al propose LLNet [18], a method to enhance noisy and darkened images. LLNet is based on the principle of stacked sparse denoising autoencoder(SSDA) for learning features to denoise signals. Low light images are synthetically generated by degrading normal light images from public datasets. The degrading pipeline consists of non-linearly darkening patches by gamma adjustment and adding Gaussian noise, both by means of MATLAB [20] functions. Multiple sets are generated by diversifying the intensities of gamma adjustments and noise levels. It is also suggested to further incorporate Poisson noise for a true sense of degradation.

2.2.2 MBLLEN

Lv et al propose a multi-branch low light enhancement model or MBLLEN [19]. MBLLEN extracts rich features from images repetitively enhance those features and finally performs multi-branch fusion to obtain the final enhanced normal light image. The model is trained on manually degraded images from the PASCAL VOC images dataset [5]. Similar to LLNet [18], images are degraded by random gamma adjustment and adding Poisson noise. The resulting images were compared to images degraded using other state-of-the-art methods to identify brightness, contrast, and other artifacts and were found to be better than BIMEF [34] and Ying [35]. Figure 2.2 shows normal light and low light images from MBLLEN training set.

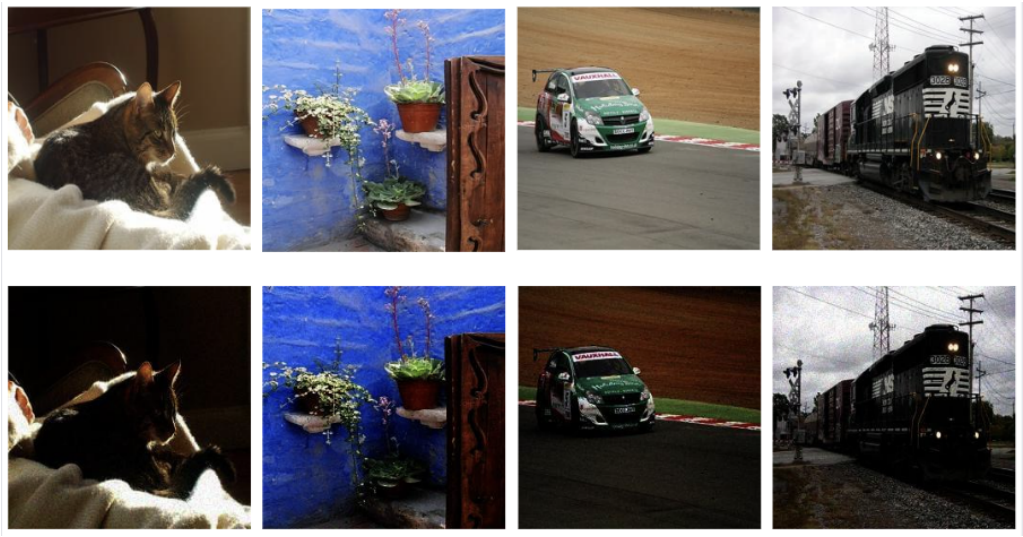


Figure 2.2: Real normal light images (above) and synthetic low light images (below) from MBLLEN dataset [21]

2.2.3 Attention Guided Low-Light Image Enhancement

Lv et al. proposed an attention-guided enhancement method using un-attention maps and noise maps in a region adaptive manner [31]. Lv et al. fabricated a synthetic dataset by identifying candidate images based on darkness estimation, blur estimation, and color estimation. 97,030 images fit this criterion out of multiple datasets comprising 344,272 images. The set of images that pass these criteria is synthesized to produce low-light images. The images were analyzed for varying degrees of exposure levels after which linear and gamma transformations are approximated for the images. These images are then verified by comparing the histograms of Y channel in YCbCr color space to that of real low light images. Further exposure fusion was used to improve color/contrast and correct exposure to avoid over-exposure in enhanced images. Finally, 22,656 images were used to train the enhancement model. Figure 2.3 shows normal light and low light images used in training.

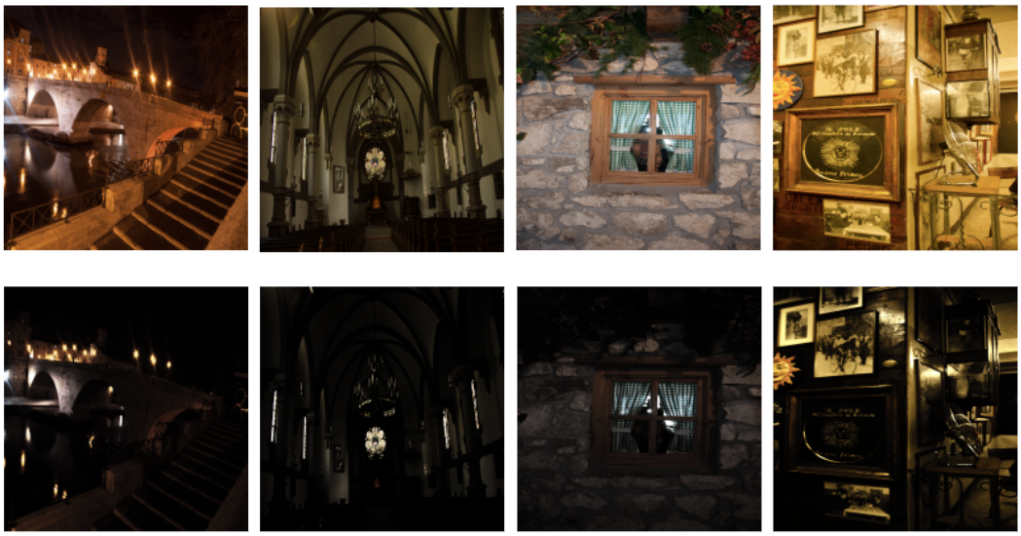


Figure 2.3: Real normal light images(above) and synthetic low light images(below) [31]

Based on the methodologies used by the degradation pipelines, low light images can be mathematically represented by $ll_{out} = n(g(nl_{in}))$, where $g()$ represents gamma adjustment and $n()$ is the inducted noise which can be gaussian noise, poisson noise or a combination of different noises.

Above mentioned degradation models/ pipelines require rigorous processing and manually estimating parameters for degradation. These parameters include minimum and maximum values for exposure which is applied across all the images. This does not account for relative brightness of different surfaces. Some degradation pipelines also fail to reproduce natural camera sensor noise and exposure which, while enhancement leads to local overexposure and noise amplification. Random gamma and Poisson noise added in the degradation pipelines can model natural noise well in normal or dim light but fails to do so in low light environments. This leads to poor characterization of structural details of real low light images.

2.3 Generative Adversarial Networks

For years now generative adversarial networks (GANs) [8] have been the at the prime solution to most common computer vision problems like super-resolution, image-to-image translation, image synthesis, image generation, etc. Generally, a GAN consists of two basic modules a generator and a discriminator. The generator is responsible for

generating fake images based on some input. The discriminator then takes this fake image and the corresponding real image and tries to identify the fake one. Thus the generator trains to create more realistic images to fool the discriminator. To ensure this learning the loss function accounts for adversarial loss. With more complex problems, better and efficient GANs have been designed for very specific problems.

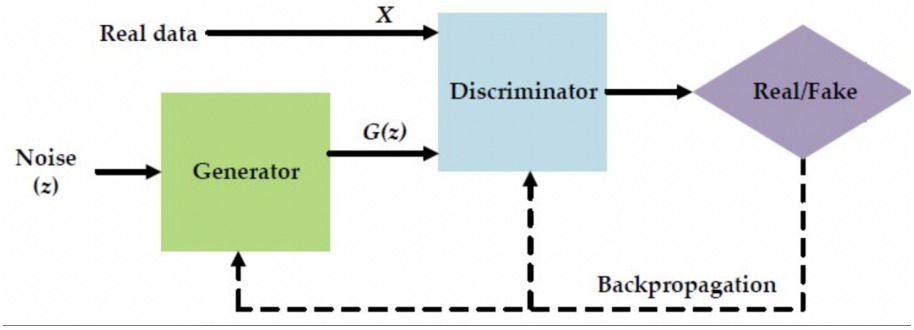


Figure 2.4: General Architecture of a GAN[7]

Condition-based image generation using GANs is also being studied actively. In some cases, class information is provided to the generator and discriminator to generate samples conditioned on it. In other cases the generator is provided with a written description or the input is appended with domain information. These cGANs have been break through in style transfer [15] and super-resolution [17].

Equation 2.1 is the standard GAN loss function as described by Goodfellow et al [8].

$$E_x[log(D(x))] + E_z[log(1 - D(G(z)))] \tag{2.1}$$

$D(x)$ is the discriminator's probability that a real instance x is real. E_x is expected value for all real data instances x. $G(z)$ is the generator's output for given input noise z. $D(G(z))$ is the discriminator's probability that a fake instance $G(z)$ is real. E_z is the expected value for all generated fake instances $G(z)$. This loss is also known as min-max loss as the generator tries to minimize it while the discriminator tries to maximize it. This loss function can further be categorized into Discriminator and Generator loss. When the discriminator is trained, it penalizes itself for misclassifying a real image as fake, or a fake image as real by maximizing the equation 2.2.

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[log\left(D\left(x^{(i)}\right)\right) + log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)\right] \tag{2.2}$$

∇_{θ_d} is the stochastic gradient of the Discriminator. m is the number of inputs.

$log(D(x))$ is the probability that the discriminator is correctly classifying the real image and maximizing $log(1 - D(G(z)))$ helps the discriminator correctly label the fake image that comes from the generator. The generator samples random noise during training and produces an output from that noise. The generator gets rewarded if it successfully fools the discriminator, and get penalized otherwise.

Equation 2.3 is minimized to train the generator:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} log\Big(1 - D\Big(G\Big(z^{(i)}\Big)\Big)\Big) \tag{2.3}$$

∇_{θ_g} is the stochastic gradient of the Generator. m is the number of inputs.

2.3.1 Challenges in GAN Training

GANs can be very useful and pretty disruptive in some areas of application but training a stable GAN is a challenging task. This is because the two neural networks, generator, and discriminator compete against each other during the training. If one of them learns faster than the other, the other network stops learning. In this case, the GAN fails to converge. The generator training can also fail because of vanishing gradients. This happens when the discriminator becomes can't pass enough information to the generator for training. This has a multiplying effect as the chain rule of differentiation and gradient starts flowing backward and it keeps on decreasing. Initial layers stop learning completely because of this.

GANs can sometimes also struggle to generalize which is known as mode collapse. Mode collapse happens when GAN generates data with a little representation of the whole dataset. It basically generates the same output for different types of input.

2.3.2 Possible Remedies

Finding solutions to mode collapse and non-convergence problems of GAN is part of active research. The convergence problem of the GAN can sometimes be solved by adding noise to the input images of the discriminator. This makes the work of the discriminator tougher and leads to stable GAN training.

Goodfellow et al [8] suggest that the min-max loss function mentioned earlier in Equation 2.1 can cause the generator to get stuck in the early stages when the discriminator

is very good at its job. They suggest to modify the generator loss such that it tries to maximize $logD(G(z))$ instead of minimizing $log(1 - D(G(z)))$. Changing the learning rate, loss function, and trying different architecture for generator and discriminator also helps make the training stable.

2.4 Image to Image Translation GANs

To establish a baseline for the solution architecture, we explore a subgroup of GANs known as image to image translation GAN that transform an image from a source domain to a target domain while maintaining content representations. There are multiple unsupervised solution for generating images like CycleGAN[37], style transfer[33] and MUNIT[11]. However, given the low light-normal light pairing readily available in RELLISUR [1], the architecture in supervised Pix2Pix [12] is a good candidate as the base for the solution. In Pix2Pix, a target image is generated conditioned on input data and the loss function changes such that the output image is feasible in the context of the target domain and is a possible translation of the input image. This can be the core of the solution where, a normal light image can be translated to a low light image. The generator in Pix2Pix is trained to both fool the discriminator and minimize the generator loss i.e. the error between the real image and the generated image thus, it can help maintain content representation.

The U-Net architecture in the generator is the same as an encoder-decoder model in the sense that it will down sample the input to a certain degree and then up sample it back to its original dimensions except that it maintains skip connections between layers of the same size in the encoder and decoder. The generator is trained using L1 loss between the real images and generated images and adversarial loss from the discriminator.

Since the model must train over multiple exposure levels, it must also learn the mapping between the exposure value and the corresponding image. StarGAN [3] performs image to image translation across multiple domains using a single generator and discriminator. The discriminator tries to identify if the translated image created by the generator is real or fake and classify it to the target domain. The discriminator is a patch GAN along with an auxiliary classifier. This auxiliary classifier can be used to learn the mapping of images and its exposure value by inducing it into the discriminator loss function.

2.5 Final Problem Statement

Based on all this, the following final problem formulation was determined:
How can a single image to image translation model convert a set of normal light images to realistic low light images with varying exposure levels having PSNR/SSIM values better than low light images generated by existing degradation pipelines?

Chapter 3

Methodology

This chapter will show different stages of implementation of our proposed model and explain the architecture used in the model.

3.1 Initial Discoveries

After analysing the existing works and the problem, we first implemented a simple supervised image to image translation GAN [12] $G : x- > y$, that will learn the mapping between a normal light image x and the corresponding low light image y of exposure value -3.0 EV by training on the RELLISUR [1] dataset. The generator G is trained to produce low light image that can not be distinguished by the discriminator D. Discriminator D is trained to detect fake images generated by the generator from the real images.

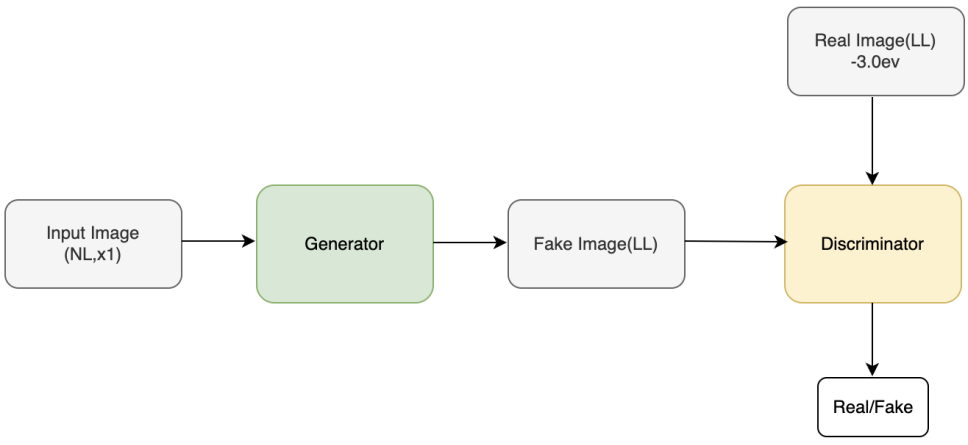


Figure 3.1: Design of GAN to generate synthetic image with exposure -3.0 EV

We designed the generator and discriminator with similar architecture as used in Pix2Pix

[12] GAN. The generator was trained using L1 loss along with the adversarial loss. After 150 epochs the model was able to generate low light images which appeared to be underexposed to the expected degree but also presented checkered artifacts. Figure 3.2 shows the output of this model. Thus, image to image translation was decided as the baseline for the final solution. There was a further need to be able to do so with multiple exposure levels and also improve on the quality of the output. Following section describes the proposed model.

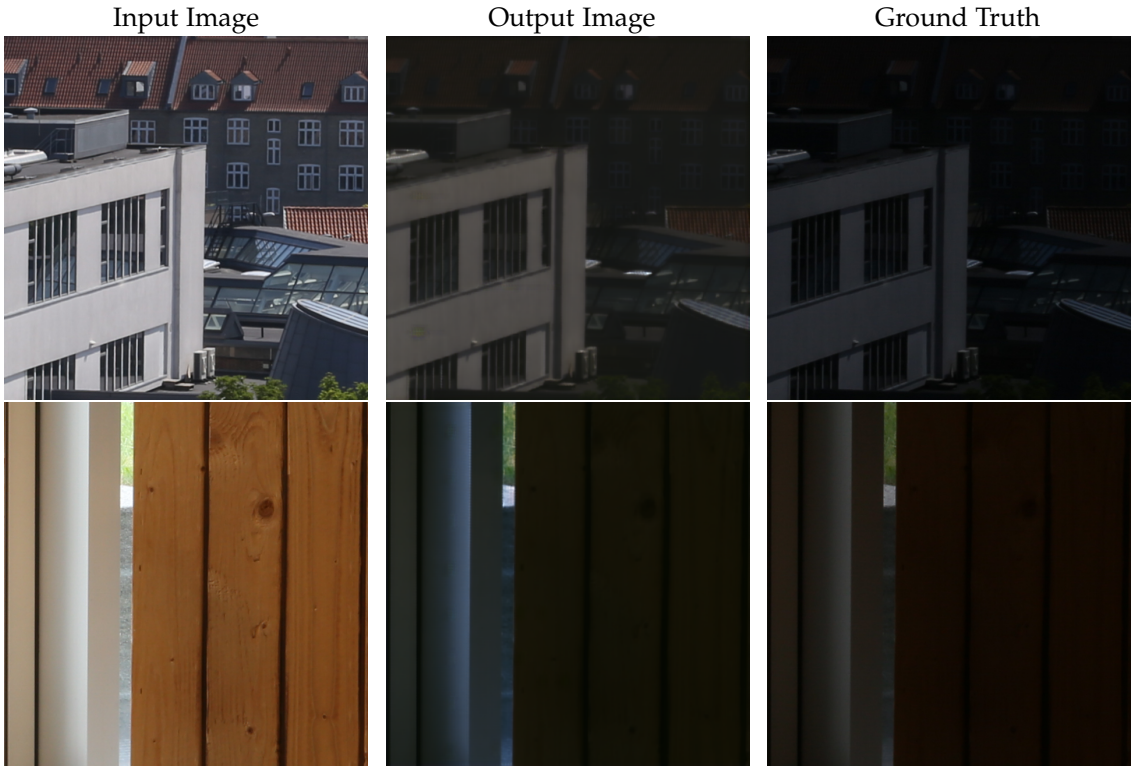


Figure 3.2: Low light images generated by a primitive GAN

3.2 Proposed model

We propose a supervised multi-domain GAN model to tackle the problems faced by the primitive model. Ledig et al used the VGG19 perceptual loss in their super-resolution model SRGAN [17] to get better perceptual quality and reduce artifacts. Thus, we chose to optimize the generator using perceptual VGG19 loss along with L1 and adversarial loss. To achieve mapping among multiple domains of exposure levels, we train the generator G to translate a normal light image x into a low light image based on the target exposure label l, $G(x, l)$ -> y. We use target label l so that G learns to map normal light image with low light images with different exposure levels that are present in the RELLISUR dataset [1]. The discriminator tries to detect fake images also predicts

the exposure value of the target/generated images. Figure 3.3 shows the described architecture.

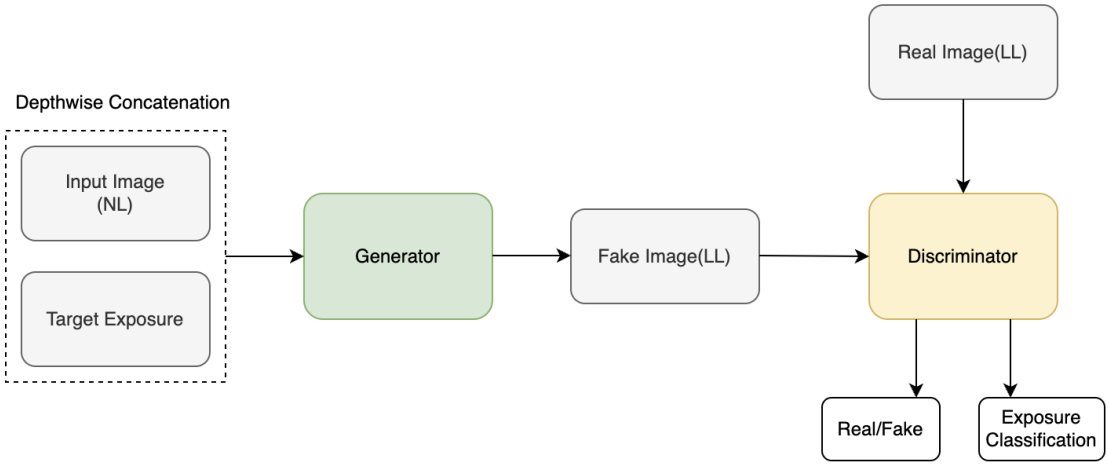


Figure 3.3: Design of the proposed model

3.2.1 Architecture of Generator and Discriminator

In the case of image to image translation problems[12][3], the input and output are different in appearance but both have the same underlying structure which means that there is a lot of low-level information that is shared between them. For example, in our case, the location of edges in the images will be the same. To make use of this low-level information, we have added a "U-Net" architecture. The U-Net structure has proven to be effective in other image to image translation GANs like Pix2Pix [12].The U-Net, first downsamples the input image, until a bottleneck layer after which it is upsampled. Skip connections are added between layer l and layer $n - l$, where n is the total number of layers. Each skip connection just concatenates all channels at layer l with those in layer $n - l$.

Our generator takes two inputs - input image and the target exposure label. We have defined target exposure labels with the help of one-hot encoding. One-hot encoding is a method used to quantify categorical data. It produces a vector with a length equal to the number of categories which is different exposure levels in our case. If the image belongs to the i^{th} category then the i^{th} bit is set to 1 and all other bits of this vector are set to 0. We spatially replicate the label vector and reshape it to (Image_size, Image_size,

number of exposure levels). We do this so that we can concatenate the target label to the input image. This concatenated input image is then fed to the U-net structure for training. Figure 3.4 shows the architecture of the generator.

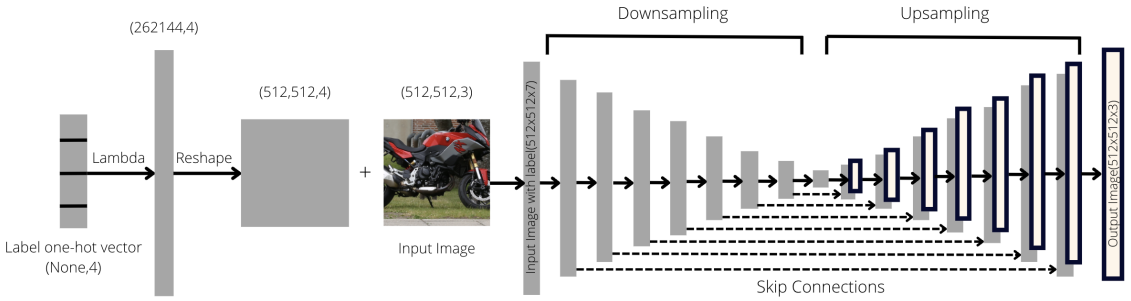


Figure 3.4: Architecture of the Generator

Figure 3.5 shows the architecture of the discriminator used in the model. Our discriminator takes the input image and target/generated image as input and predicts if the image is real or fake. It also gives the probability of that image belonging to different exposure levels in the form of a one-hot vector.

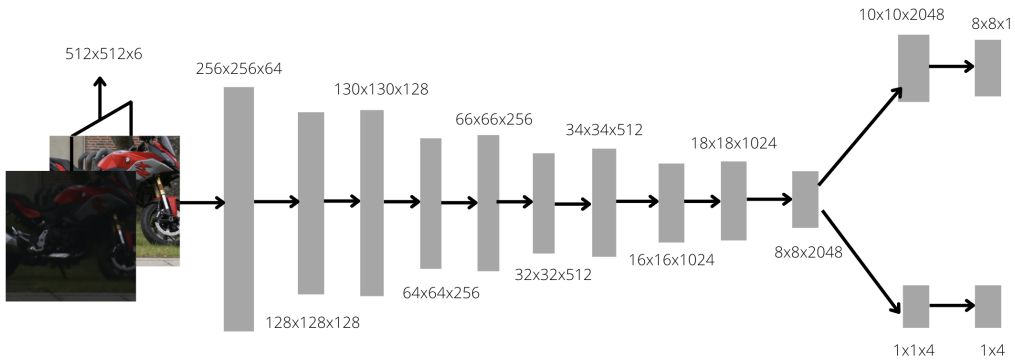


Figure 3.5: Architecture of the Discriminator

The discriminator consists of two parts- PatchGAN[12] discriminator and a classifier. We used the PatchGAN discriminator because it takes patches from the image and classifies them as real or fake. It runs convolutionally across the image and returns a single feature map of predictions that gives the final prediction. Isola et al[12] have proved that the PatchGAN discriminator promotes sharp outputs, unlike the PixelGAN discriminator which just compares pixels of the images individually. This type of discriminator can also be used on bigger images thus making it independent of the size of the image. The auxiliary classifier[23] used in the discriminator does not only distinguishes

between real and fake images but unlike conditional GAN, it predicts the class label of the generated/target image. The modules used in both generator and discriminator are of the form Convolution-BatchNorm-LeakyReLu.

3.2.2 Loss functions

Adversarial Loss

Equation 3.1 is the adversarial loss we have adopted to train the generator to make images indistinguishable from the real images.

$$\mathcal{L}_{adv} = E_x[log(D(x))] + E_{x,l}[log(1 - D(G(x,l))] \tag{3.1}$$

$G(x,l)$ is a generated image conditioned on both input image x and the target label l. D tries to detect fake images correctly by maximizing this loss whereas the generator tries to minimize it.

Exposure Classification Loss

The objective is to generate an output image y from a given input image x and a target exposure label l, which is correctly classified to the target exposure label l. We have imposed exposure classification loss when optimizing the generator and discriminator to achieve this condition. Equation 3.2 is the exposure classification loss of real images that is used to optimize discriminator during training. D_{cls} is the probability distribution over exposure labels calculated by discriminator. The discriminator tries to minimize this loss to learn to classify a real image x as its original exposure label l.

$$\mathcal{L}_{cls}^{real} = E_{x,l}[-log(D_{cls}(l|x))] \tag{3.2}$$

Equation 3.3 is the exposure classification loss of fake images that is used to optimize the generator during training. The generator tries to minimize this loss function to generate images that can be classified as the target label l.

$$\mathcal{L}_{cls}^{fake} = E_{x,l}[-log(D_{cls}(l|G(x,l))] \tag{3.3}$$

L1 Loss

We have also used the traditional L1 loss to optimize the generator. The generator is trained to generate images as close as possible to the ground truth in an L1 sense. Equation 3.4 shows how L1 loss is calculated. y is the real image and $G(x, l)$ is the image generated by the generator for input image x and target label l.

$$\mathcal{L}_{L1}(G) = E_{x,y,l}[||y - G(x, l)||_1] \tag{3.4}$$

VGG19 Perceptual Loss

We also used a perceptual loss function for generator training that depends on high-level features from a pretrained VGG19 network as mentioned by Simonyan et al[27]. This type of loss has been used previously in image processing tasks like style transfer[13] and super-resolution[17]. Perceptual loss has been found to measure similarities in images more robustly compared to per-pixel losses. Perceptual loss is the distance between the feature representations of the real image and the image generated by the generator. We used 5 layers from VGG network to calculate this feature loss. Equation 3.5 shows how perceptual loss is calculated.

$$\mathcal{L}_{perc/i,j}(G) = \frac{w_{ij}}{A_{ij}B_{ij}} \sum_{x=1}^{A_{ij}} \sum_{y=1}^{B_{ij}} ||M_{ij}(y) - M_{i,j}(G(x, l))|| \tag{3.5}$$

M_{ij} indicates the feature map obtained after j^{th} convolution before the i^{th} maxpooling layer of the VGG19 network. y is the real image and $G(x, l)$ is the generated image for input image x and target label l. A_{ij} and B_{ij} describe the dimensions of the feature maps in the network.

Full Objective

The final objective functions to optimize generator and discriminator are written, respectively, as

$$\mathcal{L}_G = \mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}_{cls}^{fake} + \lambda_{L1}\mathcal{L}_{L1} + \lambda_{per}\mathcal{L}_{perc} \tag{3.6}$$

$$\mathcal{L}_D = -\mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}_{cls}^{real} \tag{3.7}$$

λ_{L1}, λ_{cls}, λ_{per} were set to 100, 1 and 10 respectively.

3.2.3 Data Preprocessing

RELLISUR Dataset[1] was used to train the model. Out of the complete dataset, only normal light images from $x1$ and low light images from 2.5 EV, 3.0 EV, 3.5 EV, and 4.0 EV were used. We used only four exposure labels because of lack of GPU resource. For the same reason all the images were resized from 625x625 to 512x512. All the images were normalized from (0-255) to (0,1). Pairs of normal light-Low light images were created by concatenating low light images from all exposure levels to their corresponding real normal light image. Target exposure labels in the form of one hot vector were concatenated to these training pairs.

3.2.4 Training

The model was developed using Tensorflow [28] and Keras [14] in Python3 [26]. The training was conducted over a AAU Strato's[2] Nvidia T4 [22] GPU with 250 epochs. The batch size is set to 1 for instance normalization as it has proven to be efficient at image generation [12] [29]. Adam optimizer[16] was used for training with fixed learning rate for generator being $\alpha_1 = 1e-4$ and $\alpha_2 = 1e-8$ for discriminator. The learning rate for the discriminator was set slower compared to generator because with the same learning rate, the discriminator would become too good resulting in very small loss values which in turn resulted in the generator not able to learn.

Chapter 4

Results

4.1 Model Evaluation

After training the model for 250 epochs, testing was carried out on RELLISUR test dataset with low light images from -2.5 EV, -3.0 EV, -3.5 EV and -4.0 EV and normal light images from $x1$. The results are shown in Figure 4.1 and Figure 4.2.

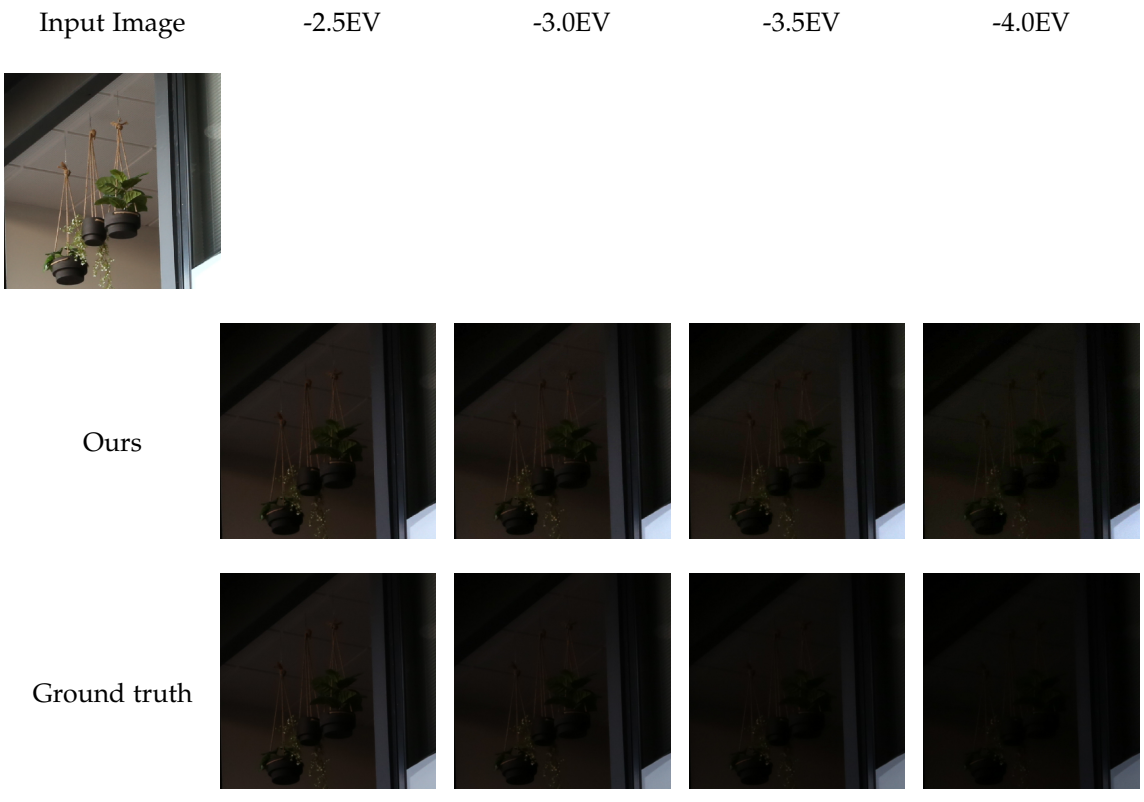


Figure 4.1: Comparison of synthetic images generated by our model with ground truth

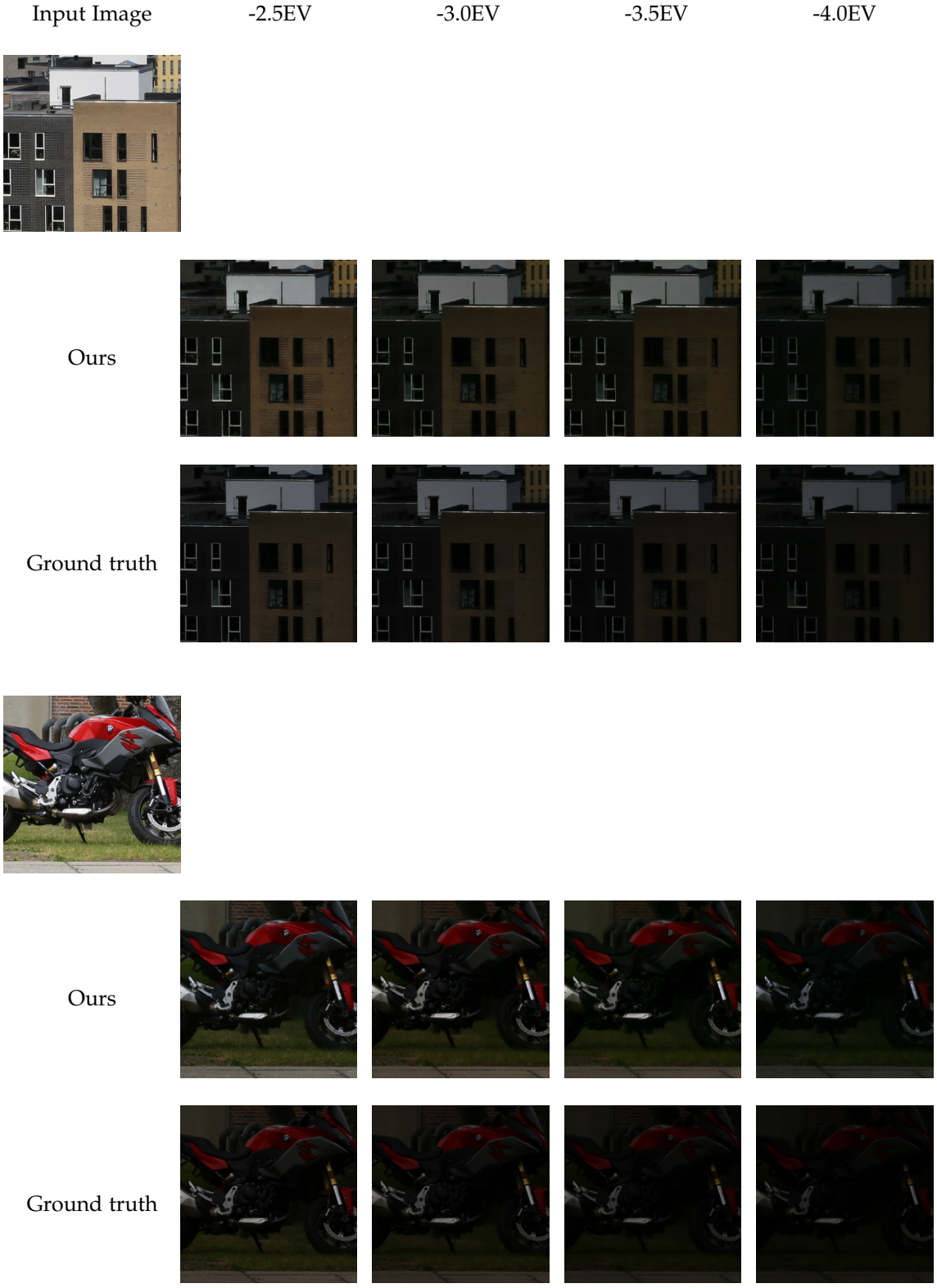


Figure 4.2: Comparison of synthetic images generated by our model with ground truth

The model is able to generate visibly underexposed images corresponding to the input EV value. The exposure also darkens with stronger exposure values.

To measure the accuracy of the model the synthetic images were compared to respective real low light images using standard measures of image similarities. PSNR and SSIM[30]

are common place measures of similarity between images. LPIPS[36] metric compare images based on factors closer to human perception and was also used to measure the accuracy of the model. LPIPS trained on Alex-Net was used for this purpose. Another measure used is DISTS[4] loss which is based on correlation of structural and texture similarity. Table 4.1 shows the result of similarity of our synthetic low light images with ground truth.

Exposure	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓
-2.5 EV	22.95	0.77	0.11	0.11
-3.0 EV	23.42	0.76	0.14	0.14
-3.5 EV	23.46	0.70	0.15	0.15
-4.0 EV	21.19	0.60	0.19	0.16

Table 4.1: Results of similarity of images with ground truth

The discriminator's classification accuracy, measured across the four levels of exposure, was based on its ability to classify the generated images according to their exposure level. Table 4.2 shows the confusion matrix of the multi-class classifier. Accuracy of the classifier was found to be 74.2%.

		Predicted			
	Exposure(EV)	-2.5	-3.0	-3.5	-4.0
Actual	-2.5	3	13	0	0
	-3.0	2	60	18	5
	-3.5	0	15	63	7
	-4.0	0	3	7	75

Table 4.2: Confusion matrix of the classifier in discriminator

4.2 Model Evaluation on LOL Dataset

To test the model on a completely different set of images to RELLISUR it was tested using the LOL [31] dataset. Normal light images from LOL were used to generate low light images with different exposure levels as seen in figure 4.3. Table 4.3 shows the comparison between the generated images and the ground truth image.

Figure 4.3: Our Model's Output on LOL dataset [31]

Exposure	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓
-2.5 EV	19.43	0.53	0.29	0.26
-3.0 EV	20.23	0.56	0.26	0.23
-3.5 EV	20.63	0.57	0.25	0.23
-4.0 EV	21.42	0.61	0.21	0.19

Table 4.3: Comparison of synthetic images generated using LOL Dataset [31] with ground truth

The gradual underexposure and degradation is coherent with the exposure values as seen in figure 4.3 and table 4.3. Based on metrics in table 4.3 images generated using -4.0 EV were closest to the ground truth images.

4.3 Comparative analysis

We compared our model to existing low light degradation pipelines described in Section 2. Normal light images from $x1$ of RELLISUR's test set were ran through the LLNET's [18] and MBLLEN's [19] degradation pipelines. Since there was no way to specify target exposure value to degrade in LLNet and MBLLEN, we approximate the exposure value by calculating the average PSNR and SSIM values of the synthetic images with ground truth of all four exposure levels used in our model. Table 4.4 shows the results of this comparison. We found them to be closest to -3.0 EV. So we chose real low light images with -3.0 EV as ground truth and used the same value as input to our model for comparison.

Exposure	LLNet [18]	MBLLEN [19]
2.5 EV	15.64dB,0.16	17.01dB,0.45
3.0 EV	15.93dB,0.17	17.24dB,0.47
3.5 EV	14.03dB,0.10	16.56dB,0.35
4.0 EV	13.91dB,0.08	16.18dB,0.34

Table 4.4: Similarity measures of degradation pipelines with ground truth of four exposure levels in RELLISUR

Table 4.5 shows the result of image similarity metrics for LLNet, MBLLEN and our model with ground truth of exposure -3.0 EV.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓
LLNet [18]	15.93	0.17	0.52	0.49
MBLLEN [19]	17.24	0.47	0.39	0.35
Ours	23.42	0.76	0.14	0.14

Table 4.5: Results of similarity of synthetic images create using different methods with ground truth of exposure -3.0 EV

Figure 4.4 shows some images from this experiment along with respective PSNR and SSIM values.

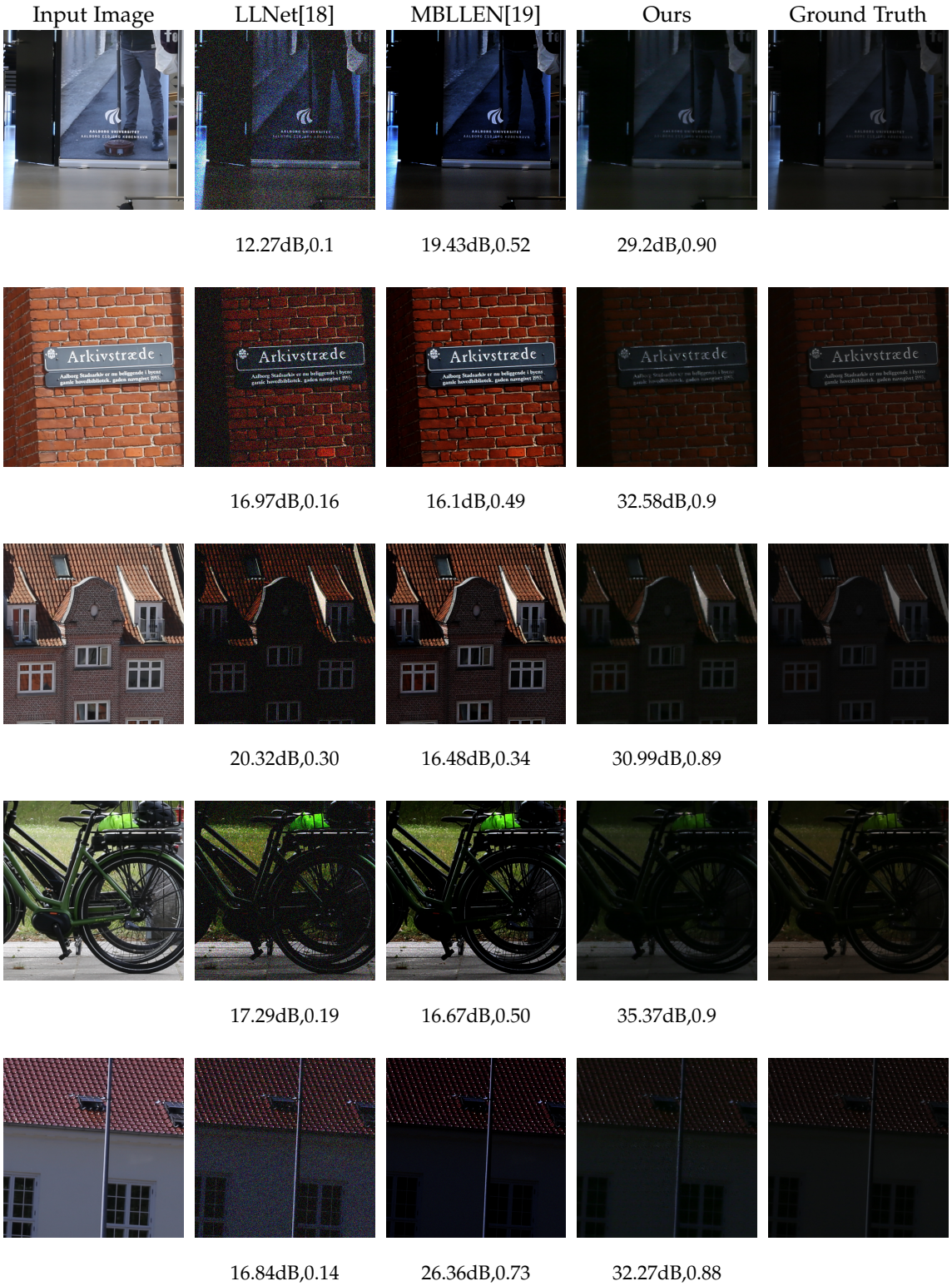
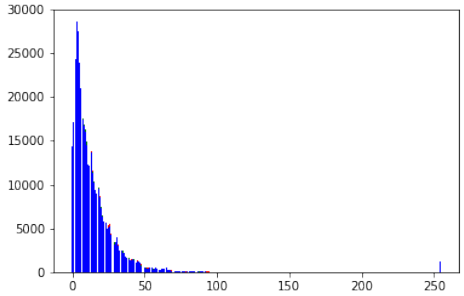


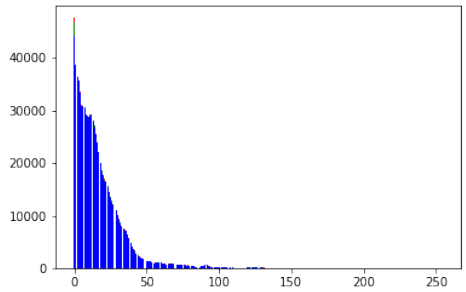
Figure 4.4: Our model's output against degradation pipelines with (PSNR,SSIM) values.

We compared histograms of real low light images in RELLISUR [1], real low light images in LOL [31] to histograms of synthetically generated low light images by our model using real normal light images in RELLISUR and those generated by degradation pipeline in MBLLEN [19] and LLNet [18]. Figure 4.5 shows how real low light images depict

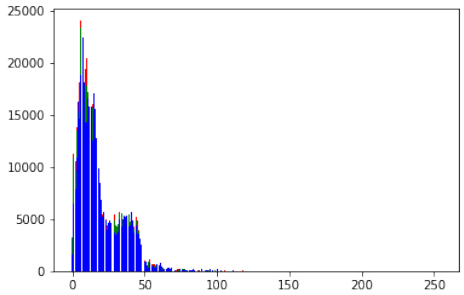
most pixels saturated towards lower values on x axis. Same is the case with the histogram for low light images generated by our model. On the contrary, the histogram for images generated by MBLLEN's and LLNet's degradation pipeline appear to be spread out across the x axis. Thus, concluding our model can generate low light images closer to real low light images.



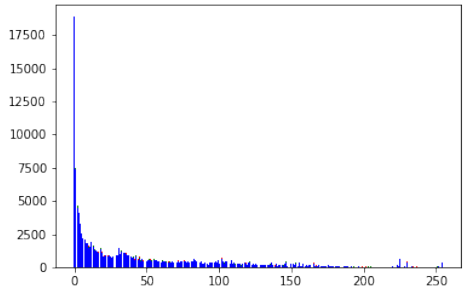
(a) Average RGB histogram of the low light images in RELLISUR[1] test dataset



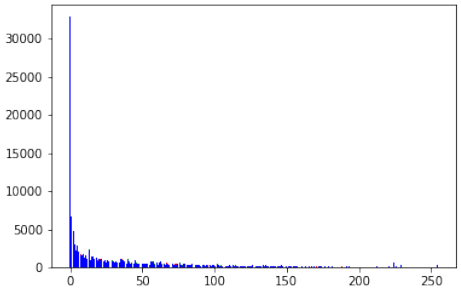
(b) Average RGB histogram of low light images in LOL[31] training dataset



(c) Average RGB histogram of synthetic low light images generated by our model



(d) Average RGB histogram of synthetic low light images generated by MBLLEN's degradation pipeline [19]



(e) Average RGB histogram of synthetic low light images generated by LLNet's degradation pipeline [18]

Figure 4.5: Histogram Comparison

4.4 Discussion

Figure 4.2 shows the output images gradually darken with higher exposure value. The PSNR value was maximum for -3.5 EV at 23.46 and minimum for -4.0 EV at 21.19. SSIM showed a different trend with -2.5 EV having the highest value 0.77 and -4.0 EV having the least at 0.60. The discriminator classifier was able to classify images with high accuracy except for those with -2.5 EV. Figure 4.3 shows the model's output to normal light images from LOL[31] dataset, which also showed gradual degradation with changing exposure values. Thus the model is found to work well with images outside of RELLISUR[1].

We compared our model to degradation pipelines used by LLNet [18] and MBLLEN [19], where we found our model to be better at creating low light images based on the PSNR and SSIM [30] values as seen in Table 4.5. We also compared histograms of real low light images to those generated by our model. There are striking similarities in the histograms where there are seldom any pixels with values greater that 50 and show a steady drop in pixel values on the x axis. However, for our model the histogram shows a bump at around 25 on the x axis which might be attributed to artifacts, glossy structures or other minor areas of improvements.

4.5 Limitations and Future Work

Even though our model is capable of synthesising low light images to a realistic degree, there are certain areas of improvements that can be explored further. Some glossy or shiny white surfaces appear to be saturated and not darken enough. Glossy surfaces also at times produce random checkerboard artifacts. Inclusion of a different perceptual loss function might help overcome this. Limited resources and other external factors couldn't let the model train on more exposure levels and had to make do with four discussed above. Due to unbalance in the RELLISUR dataset with less images of -2.5EV available, the classifier shows bias towards the other classes of exposure as seen in table 4.2. Adding weight to the less represented class might balance out this bias. Some of these issues can also be dealt by training the model longer.

To improve the model further, more exposure levels can be added to the training set to get a wider range of output images. Learning an unsupervised mapping can make

it convenient to train the model on new data that doesn't have a low light to normal light pairing. A more efficient solution than depth wise concatenation to include the exposure value for input images which creates huge memory overhead could allow for more exposure levels to be added.

Chapter 5

Conclusion

This section summarizes the answer to our final problem statement:
How can a single image to image translation model convert a set of normal light images to realistic low light images with varying exposure levels having PSNR/SSIM values better than low light images generated by existing degradation pipelines?

As proven in RELLISUR, low light image enhancement models work better with real low light images, our goal being to develop a GAN that can synthesize images that are similar to the ones in RELLISUR. We designed a GAN where the generator synthesizes low light images with exposure values ranging from -2.5 EV to -4.0 EV and the discriminator not only tries to distinguish the real low light images from the fake ones but also classifies them according to their respective exposure value. After training the model for 250 epochs on RELLISUR's training set, it was evaluated against existing degradation pipelines by means of metrics such as PSNR and SSIM and found to be better than the ones used in LLNet and MBLLEN. To verify our models viability on unseen images, the model was tested with images from LOL dataset and found to display the same qualities and degrading capabilities to those in RELLISUR. The histogram of images generated by our models was similar to the one generated by using low light images in RELLISUR.

In conclusion, we can state that our model can work to a respectable degree, coherently with RELLISUR or any other real low light image dataset to generate further low light images from normal light images with varying degrees of exposure and can bridge the gap to extend the low light image enhancement solutions across many unsolved use cases.

The implementation of the model will be available on the groups Github Page[25].

Bibliography

[1] Andreas Aakerberg, Kamal Nasrollahi, and Thomas B Moeslund. "RELLISUR: A Real Low-Light Image Super-Resolution Dataset". In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. 2021.

[2] *AAU Strato*. https://strato-new.claaudia.aau.dk/project/.

[3] Yunjey Choi et al. "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8789–8797.

[4] Keyan Ding et al. "Image Quality Assessment: Unifying Structure and Texture Similarity". In: *CoRR* abs/2004.07728 (2020). URL: https://arxiv.org/abs/2004.07728.

[5] Mark Everingham et al. "The pascal visual object classes (voc) challenge". In: *International journal of computer vision* 88.2 (2010), pp. 303–338.

[6] *Exposure triangle*. https://capturetheatlas.com/exposure-triangle-explained-photography/. Accessed: 2022-05-13.

[7] Jie Feng et al. "Generative adversarial networks based on collaborative learning and attention mechanism for hyperspectral image classification". In: *Remote Sensing* 12.7 (2020), p. 1149.

[8] Ian Goodfellow et al. *Generative adversarial nets In: Advances in Neural Information Processing Systems (NIPS)*. 2014.

[9] *How do I make my photos look awesome?* http://www.greencastlephotos.com/blog-1/2017/11/29/how-do-i-make-my-photos-look-awesome-part-iii. Accessed: 2022-05-07.

[10] *How I shoot*. https://shoottokyo.com/shoot/. Accessed: 2022-05-07.

[11] Xun Huang et al. "Multimodal unsupervised image-to-image translation". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 172–189.

[12] Phillip Isola et al. “Image-to-image translation with conditional adversarial networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134.

[13] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. “Perceptual losses for real-time style transfer and super-resolution”. In: *European conference on computer vision*. Springer. 2016, pp. 694–711.

[14] *Keras*. `https://keras.io/`.

[15] Taeksoo Kim et al. “Learning to discover cross-domain relations with generative adversarial networks”. In: *International conference on machine learning*. PMLR. 2017, pp. 1857–1865.

[16] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[17] Christian Ledig et al. “Photo-realistic single image super-resolution using a generative adversarial network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4681–4690.

[18] Kin Gwn Lore, Adedotun Akintayo, and Soumik Sarkar. “LLNet: A deep autoencoder approach to natural low-light image enhancement”. In: *Pattern Recognition* 61 (2017), pp. 650–662.

[19] Feifan Lv et al. “MBLLEN: Low-light image/video enhancement using CNNs.” In: *BMVC*. Vol. 220. 1. 2018, p. 4.

[20] *Matlab*. `https://www.mathworks.com/`.

[21] *MBLLEN Dataset*. `https://github.com/Lvfeifan/MBLLEN/issues/11`.

[22] *NVIDIA T4*. `https://www.nvidia.com/en-us/data-center/tesla-t4/`.

[23] Augustus Odena, Christopher Olah, and Jonathon Shlens. “Conditional image synthesis with auxiliary classifier gans”. In: *International conference on machine learning*. PMLR. 2017, pp. 2642–2651.

[24] *OpenCV Gamma Correction*. `https://pyimagesearch.com/2015/10/05/opencv-gamma-correction/`. Accessed: 2022-05-07.

[25] *Project Implementation Repository Link*. `https://github.com/shristy51/low-light-image-synthesis`.

[26] *Python 3*. `https://www.python.org/download/releases/3.0/`.

[27] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[28] *Tensorflow*. `https://www.tensorflow.org/`.

[29] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. “Instance normalization: The missing ingredient for fast stylization”. In: *arXiv preprint arXiv:1607.08022* (2016).

[30] Zhou Wang et al. “Image quality assessment: from error visibility to structural similarity”. In: *IEEE transactions on image processing* 13.4 (2004), pp. 600–612.

[31] Chen Wei et al. “Deep retinex decomposition for low-light enhancement”. In: *arXiv preprint arXiv:1808.04560* (2018).

[32] *White Balance*. `https://www.image-engineering.de/library/image-quality/factors/1079-white-balance`. Accessed: 2022-05-07.

[33] Zichao Yang et al. “Unsupervised text style transfer using language models as discriminators”. In: *Advances in Neural Information Processing Systems* 31 (2018).

[34] Zhenqiang Ying, Ge Li, and Wen Gao. “A bio-inspired multi-exposure fusion framework for low-light image enhancement”. In: *arXiv preprint arXiv:1711.00591* (2017).

[35] Zhenqiang Ying et al. “A new low-light image enhancement algorithm using camera response model”. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2017, pp. 3015–3022.

[36] Richard Zhang et al. “The unreasonable effectiveness of deep features as a perceptual metric”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 586–595.

[37] Jun-Yan Zhu et al. “Unpaired image-to-image translation using cycle-consistent adversarial networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232.