A Color Vision Approach for Reconstructing Color Images in Different Lighting Conditions Based on Autoencoder Technique using Deep Neural Networks

by

Paul Richie Gomes
16101284
Muhammad Arman Uddin
16101281
Hasibul Hasan Sabuj
15301123
Raian Ibn Faiz
15201026

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
April 2020

Declaration

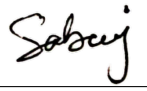
It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

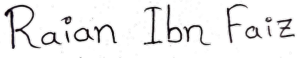
Student's Full Name & Signature:

Paul Richie Gomes
16101284

Muhammad Arman Uddin
16101281



Hasibul Hasan Sabuj
15301123



Raian Ibn Faiz
15201026

Approval

The thesis/project titled "A Color Vision Approach for Reconstructing Color Images in Different Lighting Conditions Based on Autoencoder Technique Using Deep Neural Networks" submitted by

1. Paul Richie Gomes (16101284)
2. Muhammad Arman Uddin (16101281)
3. Hasibul Hasan Sabuj (15301123)
4. Raian Ibn Faiz (15201026)

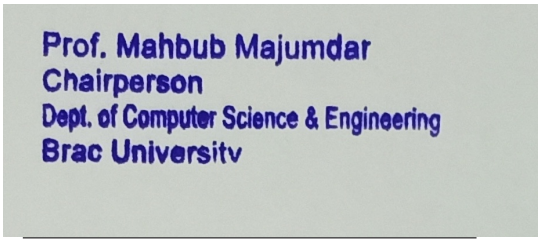
Of Spring, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on April 7, 2020.

Examining Committee:

Supervisor:
(Member)

Md. Ashraful Alam, Ph.D
Assistant Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)



Dr. Mahbubul Alam Majumder
Head of Department
Professor and Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

We propose a color vision approach that enables normalizing images based on autoencoder technique using deep neural networks. The proposed model consists of three main different steps: image processing, encoding and decoding. In the image processing part, an efficient image processing method is used to resize acquired images into a finite image resolution equal to the number of input nodes of an autoencoder. Autoencoder comprises encoding and decoding processes. Secondly, the encoding process based on deep neural networks generates a code of an input image and finally the decoding process using deep neural networks reconstructs the original image from the code generated by the encoder. The autoencoder is trained with more than ten thousand resized image dataset using convolutional neural networks. The experimental results verified that the proposed model enables reconstructing predefined normalized images from original images which can be used in sophisticated color vision applications.

Keywords: Autoencoder, Image Reconstruction, Deep Neural Networks, Color Vision.

Dedication

We want dedicate our research work firstly to our parents and almighty allah. Moreover, we want to dedicate our research to our supervisor Md. Ashraful Alam sir.

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.
Secondly, to our supervisor Md. Ashraful Alam sir for his kind support and advice in our work. He helped us whenever we needed help.
And finally to our parents without their throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AE Autoencoder

CNN Convolutional Autoencoder

$MNIST$ Modified National Institute of Standards and Technology

NIR Near-infrared

$PAIN$ Poisson autoencoder inverting network

$PICS$ Poisson Inverting Convolutions

$PSNR$ Peak signal to noise ratio

SDA Stacked Denoising Autoencoders

$UDAE$ Underwater Denoising Autoencoder

Chapter 1

Introduction

1.1 Background

One of the most influential areas of deep learning is the study of images. Images are easy to create and manage, and they are just the right type of machine learning data: easy for humans to understand, but difficult for computers. No surprise, the study of images has played a key role in the development of deep neural networks for developing such wonderful applications like detecting: vehicle, human or animal, lane, pedestrians and so on. RGB image files are traditionally and usually used in CNN training if they are available [1].These features can be used in autonomous vehicles, surveillance purposes or to observe the citizens and many other purposes. However, the problem of analyzing an image is variation of light. If the image gets sufficient light then the analysis is easier than low light. For example, in daytime the light is brighter than night because of the absence of the sun. To solve this problem different methods are currently being used for analyzing images by using both hardware and software implementations. New test to give access to precise knowledge about these perception thresholds of the colour contrasts [2]. There are various methods using artificial neural networks and image processing for classification and grading of fruits and agricultural products as reviewed in, where features such as size, shape, color, texture, etc. are used for grading [3]. Automatic harvesting it is important to detect color in outdoor condition [4].

To begin with, in 2019-2020, coronavirus (official name Covid-19) is declared as pandemic due to its highly infectious characteristics. Many developed countries like the USA, Britain found it difficult to stop spreading the virus whereas South Korea were able to control the pandemic in its country by using technology like AI for tracing contact by analyzing surveillance camera footage and quarantine them immediately [5], [6]. Also, many developed countries use artificial intelligence over the surveillance camera to measure the speed of the vehicles in the highway and detect the vehicle number right away. It helps them to punish the suspected driver almost hundred percent accurately and to ensure road safety as well [7]. Moreover, early detection of disease such as Parkinson's disease can be also be done using ANN and other image processing techniques [8].

However, there are several methods used for detecting an object like R-CNN, YOLO V2/V3 etc. The problem arises when the sunlight turns dark. Detecting objects

in the dark is a bit difficult for humans as well as the machine. As the machine is not intelligent as human, it has to be trained or has to be equipped with the technology to overcome this barrier. As a result, night vision cameras, grayscale images etc. are used for overcoming the low light image analysis problem. In other words, enhancement of images amplifies visible light. That makes it easier to see the pictures. Tiny pieces of light are visible even at the darkest hours. All of this light may be infrared light that can't be seen by humans. Night vision goggles capture all available light using Image Enhancement technology. Instead, they intensify it so you can actually see what's happening in the night [9].

However, there are several methods used for detecting an object like R-CNN, YOLO V2/V3 etc. The problem arises when the sunlight turns dark. Detecting objects in the dark is a bit difficult for humans as well as the machine. As the machine is not intelligent as human, it has to be trained or has to be equipped with the technology to overcome this barrier. Color histogram is one of the important and widely used features to describe the content of an image [10].The major issues in object detection are the shape variance, lighting variance and objects' pose variances [11].The feed forward neural network is implemented to estimate the transformation, defined in terms of the translation, rotation and magnification parameters, to align the corresponding images [12].

As a result, night vision cameras, gray scale images etc. are used for overcoming the low light image analysis problem. In other words, enhancement of images amplifies visible light. That makes it easier to see the pictures. It is a challenge to display the captured scene in an authentic color image [13].Tiny pieces of light are visible even at the darkest hours. All of this light may be infrared light that can't be seen by humans. Night vision goggles capture all available light using Image Enhancement technology. Instead, they intensify it so you can actually see what's happening in the night [9].

Furthermore, the alternative is using what is called thermal imaging. Instead of searching for the light that things are reflecting, then we are searching for the heat that they give off. Generally speaking, living objects traveling in the dark would be hotter than their surroundings; this goes for cars and machines as well. Hot objects give off infrared radiation, a light-like source of energy but with a slightly longer wavelength (lower frequency). Creating a camera that absorbs infrared radiation and converts it into visible light is fairly easy: This operates like a digital camera except that an image detector chip (i.e., a load-coupled device (CCD) or a CMOS image sensor) responds to infrared rather than visible light; it still generates a transparent image on a screen in the same way as a typical digital camera [14].

On the other hand, the acquired photos are distinguished by the presence of noises under the low light conditions. This induces not only the loss of visual perception of the image but also the decrease in the rate of detection of objects. Denoising an image is normally connected to filtering activity. A variety of known methods of denouncing images exist, such as: Gaussian smoothing model [15], Bilateral filter [16].Moreover, noises such as speckle and white noise are reduced in hologram reconstruction process[17]. More noise reduction method includes Zhou-Wang wavelet

total variation [18], etc.

Therefore, it is clearly understood that analyzing an image in various conditions needs various approaches due to lights.Sparseness of light tends to move in more complex analysis. This research however mainly focuses on intensity of the light by using an AE which is a kind of deep neural network. That is, any image which tends to have low light or nearly no light then the model converts the image such that it fills with full of light like daylight image.

1.2 Research Problem

In this thesis we proposed and demonstrated a color vision approach using deep neural networks based on autoencoder technique that allows converting real world images into normalized images to be applied in advanced color vision/color machine vision applications such as robot vision and other industrial applications.

1.3 Research Objective

The purpose of this research work is to eradicate efforts to do for darkness or low light. The main goal of this research is to convert an image brighter which can be found dark or lacking light. The objectives of this research work are:

- To prepare a dataset of images in various lighting conditions.
- To reconstruct an image through a deep neural network.
- To study performance of the proposed and state of the art models.

1.4 Scope and Limitations

Through this thesis paper it has been tried to develop a system which can reconstruct an image entirely by variation of its color. Reconstruction of image may play a vital role in various aspects. Many models, algorithms are used to overcome the low light image configuration. By using this model one can find it useful for not doing extra hard work. Instead of doing extra hard work of detecting an object from an image using various models, algorithm for low lighting one can concentrate on other tasks. However, it is difficult to reverse the work, that is, this model cannot make a bright image to dark which might be necessary sometimes. For instance, to predict the gloomy weather how it looks or to portrait an image of how it seems to watch in the morning or evening. Besides, predicting a dark image is difficult when the moon shines at night because the night looks a bit brighter than a usual dark night.

1.5 Thesis Report Outline

The rest of the portion of this research paper contains the following. Chapter 2 contains the background study and literature review. It describes the existing works on

this topic and different types of machine learning algorithms. Chapter 3 is methodology which describes the structure of our total works. It shows the whole process of our proposed model with proper diagrams. This portion will contain the information on the dataset, data preprocessing and feature extraction and classification model. In chapter 4, results and analysis of our model is discussed. It describes about the results with confusion matrix and classification reports. This portion will show the parameters to find out the comparison of our result and also the finding from these results. Chapter 5 concludes the research work by summarizing our whole work.

Chapter 2

Literature Review

Computer vision deals with the problems that general algorithms fail to determine.Computer visionis a subfield of artificial intelligence (AI) aimed at understanding still images and video sequences [19].Human eyes are extremely sensitive to light as we can differentiate various colors. Color constancy models and with the help of computer graphics models the system can achieve better results compared to human vision systems.Many challenges still exist in computer vision research area because of the limited information provided by RGB images [20].Images and videos are usually encoded by lossy compression methods due to their large memory or bandwidth requirement [21] However, the retinal system works in a way where color can vary from person to person in perspective [22]. Computer vision will help us achieve constant success across the domain. The objective of our research lies in proper color variation determination color illuminance determination with proper co-efficient with reflectance models and CNN architecture. Pretrained CNN can extract better features for input images and can reconstruct image for better feature visualization [23].Color discrimination is the most important factor in object detection and recognition it should be observed in advance computer vision test [24].Computers have to process visual information in data space formed by the robustly detectable but less meaningful features such as colors, textures etc [25]. Image reconstruction sometimes becomes a concern for object detection. Stereoscopic techniques are used for better image reconstruction in 3D images [26]. General images can be reconstructed using general neural networks.

Human vision of color recognizing and detecting object can be accumulated in the concept of computer vision system for autonomous vehicles. However, images captured by a camera are influenced by the chromaticity of illumination [27]. Object recognition plays a vital role in machine learning, computer vision, and robotics in the modern technologies [28] Color constancy processing can help as a pre-processing step to make sure that recorded color of the objects in the scene does not change under different illumination conditions [29].Various weather conditions such as haze, fog, smoke, rain will affect the scene analysis and disrupt classification [22]. Datasets of night and make better use of RGB matrix and color constancy. Red , green and blue mixture can be seen by human eye [30]. Various physical characteristics such as illumination, orientation, depth, reflectance, velocity are the parameters upon which an image of a scene depends on [22]. In this research paper [31], the researchers proposed a method for reducing noise from images with the assistance of

sparse AE. These enthusiasts have done a number of studies to test the efficiency of the approaches in different scenarios. These authors selected gray images and gray versions of datasets as their experimental sources and they trained their network with various datasets. The use of neural networks in image processing has conventionally been limited to its use in image compression or image restoration [32].

In addition, they included multiple noises with various criterions in a significant number of images in their datasets. In order to fit in various situations they experimented on images with multiple noises in a single image. They trained their model by testing their datasets included with various noises such as Gaussian noise. In this paper, they also included Salt noise and Speckle noise in their datasets along with Pepper noise for comparison. However, after they employed it to a testing set under another kind of noise that is different from the sort they did not prepare their model, the function started to collapse. However, they discovered that the Hybrid image datasets can generate more successful outcomes. The performance is estimated in terms of peak signal to noise ratio (PSNR) [33] that is a widely used quality index. The hybrid noise training sets provide a higher value of the PSNR index absolutely on the maximum of other networks. These researchers found very strong results for various situations by their method using hybrid datasets to train the sparse denoising AE. In this important paper [34], these authors presented a dual AE network model to reduce the noise from images and intensify the low light condition of images. These researchers merged the convolutional and stacked AEs to create the paired AE network model and they did their research with the idea of retinex theory [35]. The principal objective of their designed network model is to achieve noise reduction and brightness enhancement. In their research, on the illumination part the stacked AE can be considered as the brightness constraint or smoothness term. These enthusiasts used convolutional AE to apply the penalty on the reflectance element to diminish the artificial noise. Their proposed model used a set of regions of both high and low contrast images to train the neural network. The stacked denoising autoencoder can be applied to remove noise from images. On the other hand, the convolutional autoencoder can train by using two-dimensional structural information. It eliminates the lack of details. Their proposed dual AE can therefore calculate the raised reflectance with the assistance of convolutionary AE to minimize the expansion of noise. Since the reduced portion of incoming light brings the low lighting portion, the over-enhanced reflectance component is very likely to be obtained since the predicted reflectance is inversely proportional to the illumination element. They compared their revealed paired AE model to avoid overriding of the reflectance components with the drawbacks of the variational framework of retinex method in this paper. Their proposed method can minimize the augmented noise in the intensified reflectance component by using the convolutional AE for noise reduction. Moreover, their proposed network model can generate the output equivalent to input in a self-monitored way of learning that can be achieved by training their model. Since there is the possibility of obvious difficulty to acquire the pairs of low and high contrast images they created a collection of data for experiment supported by pairs of low and high contrast pixels. These researchers proposed a design that can generate strong results with the assistance of a dual AE model focused on the theory of retinex without augmentation and saturation. These researchers proposed

this model which can produce the high-quality image in low light conditions for many applications of image processing for instance visual monitoring system (VMS).

In this paper [36] these researchers worked on UDAE. They tried to maintain the accuracy and estimated cost to allow real time implementation applying the AE network on their proposed model. They found it difficult to gather data collection to train the network as it is difficult to get clear images. They obtained data from large fish aquariums, different pictures taken from software-processed video and underwater images that were captured in a nearby distance to artificial light exposed structure. These authors collected more than fifteen thousand see-through and distorted images where 7,055 pairs of images were produced and filtered. They used the CycleGAN generative model [37] for style transferring of images. The CycleGAN model training took more than one week on 4 NVIDIA TITAN X GPU devices. After removing the unsuccessful cases 5,194 pairs of images remained. The limitations of CycleGAN in style transferring was the reason behind the failure cases. These enthusiasts restored the color of the underwater image applying a single denoising AE. Their UDAE's testing on NVIDIA Quadro M5000 took about one day. They instructed and trained their proposed network model to restore the color of underwater images. UDAE network was trained to evaluate its simplification capability on real data for instance underwater videos extracted from social media. These researchers showed that underwater images can be reconstructed using a single denoising AE based network and it is easier to use a single AE for real time implementation. In addition, their network model showed that UDAE can generate better results for color restoration in images.

In another research paper [38], these authors researched several deep learning architectures that can be applicable for compressed sensing. They explored their experiment in the area of photon-limited imaging. These researchers applied a stacked AE and convolutional neural network to implement signal reconstruction. They designed their proposed model to reconstruct compressed signals assessed with Poisson noise applying PAIN architecture. They used MNIST dataset to train their model and uphold deep neural networks following reconstruction methods. In their proposed model they applied three different deep learning architectures to restore data from noisy low dimensional images. These authors trained their all three architectures by using the backpropagation algorithm and they used mean squared error (MSE) as a loss function. The three architectures they suggested were SDA, PICS and PAIN. The MNIST dataset has seventy thousand images of handwritten numbers where they took sixty thousand images for training and took others for testing. The PAIN and PICS architectures generated far better results than SDA in the area of compression. They found that PAIN architecture can surprisingly regenerate initial images with higher intensity. Their experiment showed that the MSE often delivers a lower MSE value for all three architectures. The proposed model showed that PICS architecture is more exquisite in figures and while the compression raises, SDA and PAIN architectures are more resemblance to the PICS architecture. They discovered that considering overall performance PAIN and PICS networks can generate better results using MSE as a resulting metric compared to SDAE.

In this relevant research paper [39], these enthusiasts proposed a method for re-

constructing a color image sequence and reducing the noise along with blurriness from the captured color images. They used near-infrared (NIR) images which are captured in very low brightness conditions with different time of exposure for testing their network model. In this paper, they presented a system that can capture the RGB and the NIR images at the same time. They used an RGB camera and NIR camera in their imaging system to capture RGB/NIR images. The long exposure image of an RGB color model can receive adequate color information. On the other hand the short exposure NIR image can capture timely changes in the background. Therefore, their imaging system can gather required information of the image sequence for reconstruction. These researchers trained their algorithm with artificial images as well as real images. In addition, they made comparisons with guided image filtering [40] and Yan's method [41] that minimize the noise in color images applying NIR images in their research paper. They received significant results from this comparison. Their research showed that their algorithm has capability to reduce excessive noise from color images and reconstruct the images where other methods did not succeed. They examined their algorithm with the images with low light background and excessive low light background with long exposure time. Their experimental result showed that they resolve the critical problem created by the very dark atmosphere. There are some drawbacks also in their imaging system. One of the challenges is to suppress the haze in motion and excessive noise completely. They suggested selecting optimal control parameters (k) figuring on the background of the scene. Moreover, they showed better performance by manually installing the parameters for places where major motion exists. One more drawback of their network model is that it utilizes two cameras in order to adjust the pixel spot between the cameras correctly. If there is a slight difference between the cameras, their system may struggle to produce appropriate results. These authors pointed out that using one camera for capturing both RGB and NIR images can be the approach to overcome this drawback.

Chapter 3

Proposed Methodology

In order to develop a proper efficient reconstruction technique in different color variations for images, We took pictures in different directions. There were many different types of objects in these different directions, we worked on the color differences of those objects. The color of the object varies at different times of the day. To produce more color variation, five different directions are used. By training these color variations in the autoencoder, the encoding part resizes the image according to its model and refers to the bottleneck of the autoencoder. Decoding part then takes the resized image as input from the bottleneck and produces a normalized image. Each layer uses different activation functions which are ReLu and Sigmoid to validate the image as output in the corresponding layer. In the following sections the working flow of the autoencoder is discussed consecutively.

3.1 Structure of Proposed Model

Figure 3.1 illustrates the proposed model. The proposed model consists of two main techniques: image processing and autoencoder. The image processing includes image resizing and color space conversion. The autoencoder is composed of two different processes: encoding and decoding as shown in Figure 3.1.

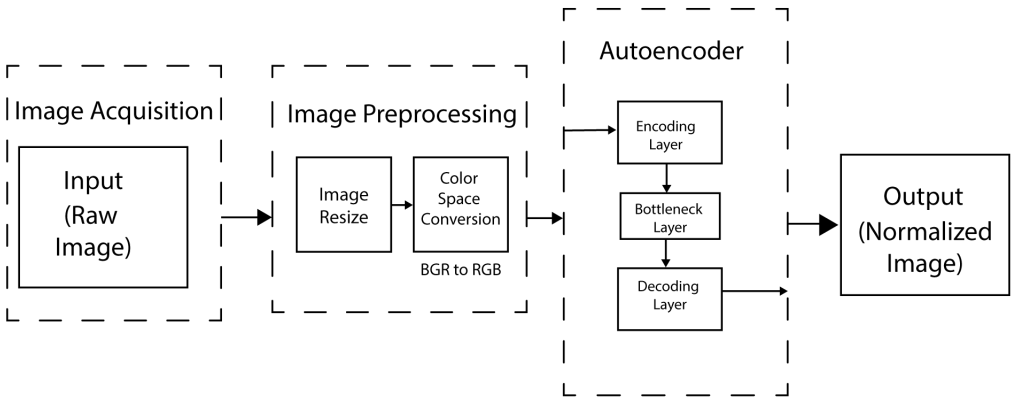


Figure 3.1: Structure of proposed model

3.2 Image Acquisition

Image acquisition defines the first part of the model where the image dataset is collected in raw image format with a pixel value of 1920 x 1080 pixel. For the research purpose a unique dataset has been generated. Generally, it is very difficult to accumulate stable images for a scenario where we can speculate the same image for different lighting conditions. Moreover, open source datasets contain various anomalies such as lack of ambient lighting, proper angle variation, time definition and pure RGB distribution. As time dimension, sunlight and defined angle distribution was important in order for our model to be trained and tested properly based on our autoencoder structure. Therefore, creating a unique dataset provided an opportunity to properly define the problem domain and work accordingly. Our dataset consists of 10,000 images shot in 5 different angles with equal distribution of 1000 in each angle. The dataset is divided and collected based on 5 different positions. A device is used which is xiaomi yi lite 4k camera to capture the images. This camera can produce images with wide angles, sharp images and provide eye-view images. On the day of the collection of the dataset the highest recorded temperature was 31 degree Celsius and lowest recorded temperature was 23 degrees Celsius in that particular zone. Lighting conditions and temperature are correlated weather conditions. Therefore, temperature plays an important role in dataset collection as sun and cloud can affect the amount of light that can be refracted and reflected from the image to be captured on the image. Moreover, time and lighting condition are also correlated factors. In the context of Bangladesh, this dataset was collected from 9 am to 4 pm. The timing dimension affects our dataset collection as it will be difficult to train the initial dataset if they are collected before sunrise and after sunset. The images are in 1920x1080 pixel (2073600 pixel) format. Wind conditions can add a lot of noise in an event of collection of an image. Wind conditions during the collection of the dataset were stable. Unnecessary objects that have a lot of vibrant colour was avoided as it could potentially give us a color value which can cause a spike on RGB histogram resulting in dominance of one colour over the training time-frame. And this could also lead to better training in one colour dimension and under performing results in another colour dimension. The dataset is labelled into two categories afternoon and noon. Each of the categories consists of 5 different angles to speculate the difference of lighting condition. Our dataset consists of 1000 images per angle. The model has made a validation split of 10% and trained 80% images and used 10% images as our test images. This distribution was made to avoid overfitting of data as RGB values can overlap during the AE training.

3.3 Image Preprocessing

Due to hardware limitations our images are resized into two different pixel dimensions 148 x 196 and 380 x 420 pixel size where we have mainly shown the results for 380 x 420 pixel to validate our theoretical approach. After the initial resizing of the image later part contains the image color space transformation of BGR to RGB. This approach is taken in this model due to the image importing mechanism of OpenCv. OpenCV generally imports images in BGR format due to its unique importing feature. However, our problem domain mainly involves the computation on RGB features. Therefore, we have opted to use the inbuilt feature of OpenCv

architecture to convert to RGB color space. During this process it is important to make sure during the conversion that our image values do not change. Therefore, it was checked for every image if the conversion was made successfully as the conversion process only transfers values of one array to another.

3.4 Autoencoder

An autoencoder is a technique which attempts to rebuild its output as much as close to its input. It has mainly two parts such as encoding and decoding. Figure 3.2 represents the detailed workflow of the autoencoder and its processing in our model.

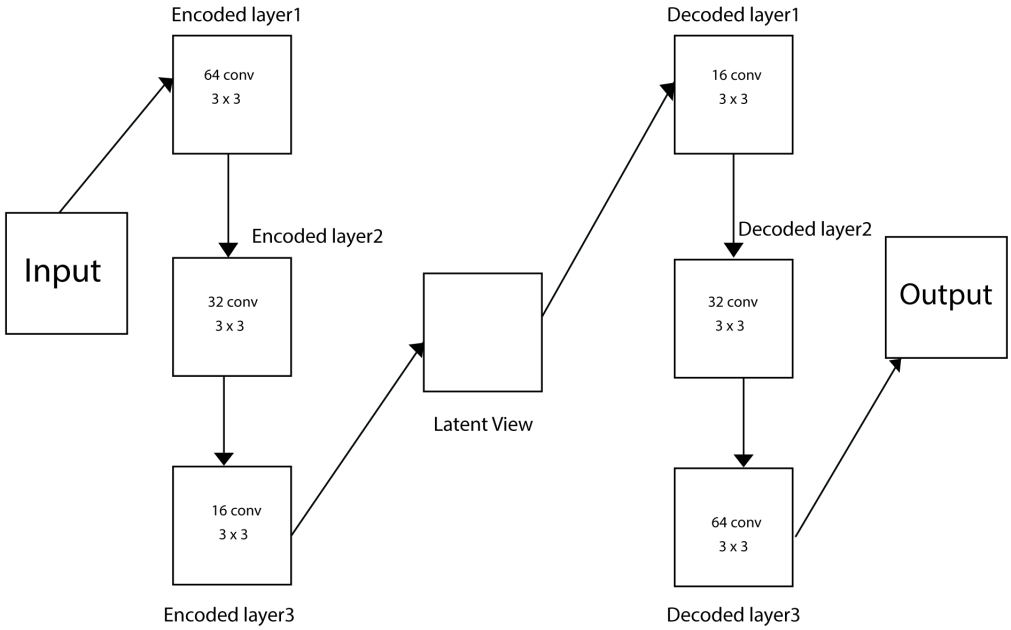


Figure 3.2: Workflow of autoencoder

3.4.1 Encoding layer

Figure 3.3 shows a part of an AE where it learns how to reduce the input dimensions and converts the input into lower-dimensional code gradually.Here, encoding layer works by encoding multiple or single layer encoding layers which are also basically known as convolution layers. Encoding layer consists of three encoding convolutional layers. Our model works by running max pooling layer after each encoding layer. Third encoding layer sends the output of the third layer to the bottleneck layer. This part of the model compresses the image by taking the significant features of our image.In our encoding architecture convolution layer is used to learn colour drops and colour value of an input image. We have used three encoding layers in our encoding architecture. In our first encoding convolution layer we have used 64 filters with a weight matrix of 3x3 dimension. In our second encoded layer 32 filters are used with a weight matrix of 3x3 dimension. In our third layer 16 filters with 3x3 dimension weight matrix are used. The third maxpool layer is our latent view which is the final layer before the decoding layer starts.

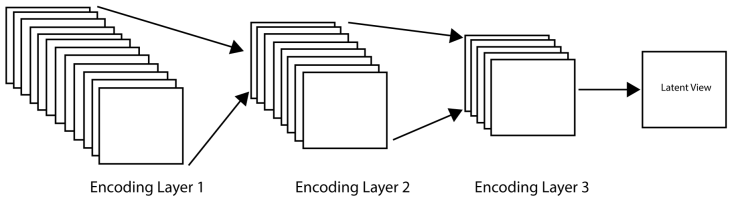


Figure 3.3: Architecture of encoding in proposed model

3.4.2 Bottleneck layer

Bottleneck layer also known as latent view layer represents the final reduced dimension of the input image which was provided in the input part of the image. In our bottleneck layer represents three hidden layers. Due to the complexity of our training we have used three hidden layers as it could provide more depth to our final output. RGB image is quite a complex domain as it has its own 256x256x256 pixels which represents approximately 16.8 million colour combinations. As convolutional layers operate on feature maps with two spatial axes which are height, width and depth. For our problem domain we have used 3 as the dimension of depth axis because our images and our problem domain mainly focuses on red, green and blue images. Therefore, our output layer is 3D form tensor.

3.4.3 Decoding layer

Decoded layer works by performing the final stages of convolution operation. Figure 3.4 represents, the image's dimensionality increases and the image is resized to the original input image. Upsampling is used to increase the image dimensions with learnt features everytime the model runs. The Decoding layer consists of three decoding layers. Third layer's output determines the final output which is the normalized image in this model. In decoding architecture, we have used three decoding layers. In the first convolution layer we have used 16 filters with a weight matrix of 3x3. Second and third convolutional layer we have used 32 and 64 filters respectively with a weight matrix of 3x3. In our decoding architecture we have used up sampling by using a 2x2 filter after convolution layer. Stride 2 is used to maintain the ratio of max pooling. It helps to reconstruct images.

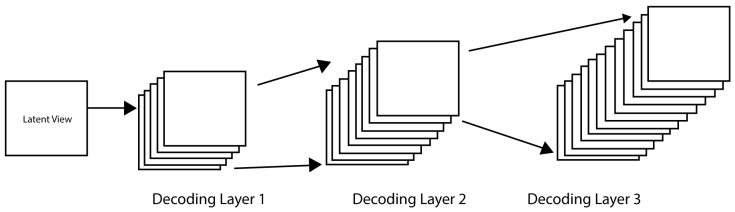


Figure 3.4: Architecture of decoding in proposed model

Output: The output of the suggested model depicts the output image in size of 380 x 420 pixel. This output is in RGB format. The resultant output that we obtain is of the opposite timeframe than the input timeframe. For instance, if we give an image of an afternoon lighting condition, our model provides output for an image of a noon

lighting condition. The model provides output for corresponding angles.RGB values can change depending on color of lighting condition as color space is a primary factor in learning the outcome of our model. Lighting condition is a primary concern for the image to produce an effective output. RGB values can change depending on color of lighting condition as color space is a primary factor in learning the outcome of our model. Therefore, a normalized image is obtained in our model.

This proposed model is defined by an encoding and decoding architecture with several convolution layers. Our model uses a structure of AE which followed the rigorous training and testing of our own dataset. Our model can generate two kinds of output depending on the input. One output defines the afternoon lighting condition and another output defines the noon lighting condition. 2D maxpool is used in our model to sample the input image so that assumptions of feature can be made conveniently. We have used a 2x2 filter that will be used upon the input layer after every convolution layer stride value 2 is used for smooth movement from images. Max Pooling will help our model in better extraction of sharp and smooth features. Figure 3.5 depicts the pooling layer structure for the third encoded layer and first decoded layer.

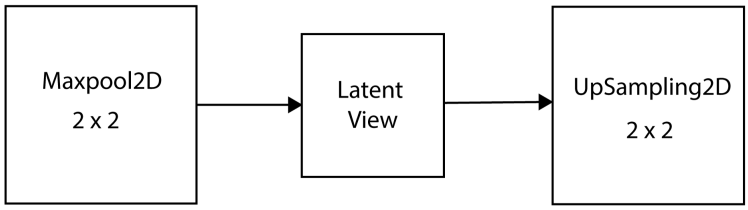


Figure 3.5: Maxpool and upsampling architecture

3.5 Activation Functions

The proposed model has used Relu (Rectifier Linear Unit) activation function in all the layers except the output layer where this model has used sigmoid function. Relu is used in processing part of our model as it helps this model for interaction effects and non-linear effects. Relu in accordance with the bias term makes this model more efficient as it is a monotonic function. It makes this model faster for taking less time to train and run. Vanishing gradient problem is not faced by the ReLu activation function. It converges faster than other functions. We didn't have to rely on leaky ReLu as there is a lack of negative value in the RGB domain and it would leave a detrimental effect on our model. As we have no negative value in the RGB domain its sparse feature will make it by ignoring for a false negative value. ReLu is mathematically defined as y=max(0,x) [42]. The Figure 3.6 depicts the ReLu function graph.

Sigmoid function is used in the output layer of our architecture to avoid unnecessary changes of output over minor changes to input and integrity of our architecture. Formula for sigmoid function. Having only to work on the color features of our model in RGB format and basic underlying features we can get a good grasp on our

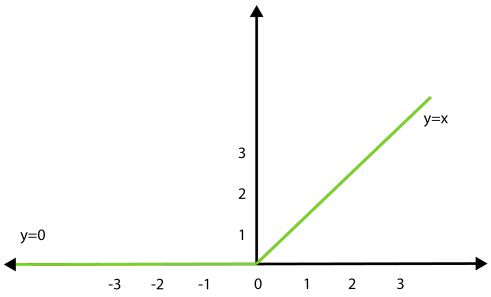


Figure 3.6: Graph of ReLu activation function

output due to sigmoid function. If a value is far from the original value sigmoid function makes it possible to relate that value to the original value to reduce the deviation between the input and output values. This feature in contrast to our AE model works well to maintain the features of our original images while training and validation split. After the successful training of model results are well distributed for the use of sigmoid activation function $Y = \frac{1}{1+e^{-x}}$ [43]. The Figure 3.7 depicts the sigmoid function graph.

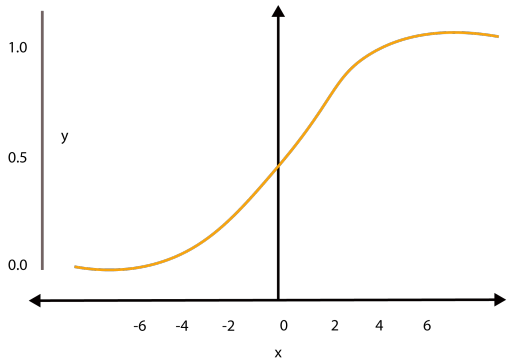


Figure 3.7: Graph of sigmoid activation function

3.6 Adam Optimizer and Cross Validation

Adam (Adaptive Moment Estimation) optimizer is used as it is a stable optimization algorithm for computer vision image processing. It is used in our model to update network weights iteratively based on training data. Moreover, Adam optimizer used both the benefits of RMSProp and Momentum. Here, updating of parameters is invariant to rescaling the gradient. Stationary objective is not required by Adam Optimizer. Annealing of step size is done naturally. For epoch we have used two different epochs to determine whether variation of epoch creates a difference in training and validation of our data. We have used standard validation of .1 to properly fine tune the model hyper parameters. Therefore, cross validation is used in our model to avoid overfitting and underfitting. Furthermore, we have used 20 as our batch size. This descent hyper parameter defines the number of samples that our model works through before updating the internal model parameters.

3.7 Dataset

For this research purpose an unique dataset has been created. Generally, it is very difficult to accumulate stable images for a scenario where we can speculate the same image for different lighting conditions. Moreover, open source datasets contain various anomalies such as lack of ambient lighting, proper angle variation, time definition and pure RGB distribution. As time dimension, sunlight and defined angle distribution was important in order for our model to be trained and tested properly based on our autoencoder structure. Therefore, creating a unique dataset provided an opportunity to properly define our problem domain and work accordingly. The dataset consists of 10,000 images shot in 5 different angles with equal distribution of 1000 in each angle.

3.7.1 Data Cleaning

To make a clean dataset of clear images of stable RGB distribution blurry images, unusual noise distortion images with moving objects such as birds and humans were removed. The images which were out of focus were removed to maintain the integrity of our training and testing methodology. Moreover, images with almost similar RGB values were cleared of our dataset to avoid training of similar images more than one time. Further, more than 10% images of our initial collection with overexposed ISO were removed from our dataset. Excessive zooming of background was avoided to maintain 100% of original images to avoid import loss.

3.7.2 RGB Color Space Distribution

In the model dataset as there are two time zones that have been chosen for this research purpose, this methodology provides the RGB distribution for the respective images. In Figure 3.8 section (a) and section (b), it has shown the RGB histogram for angle 5(W) sample image of afternoon time frame. In section (c) and section (d), it has shown RGB Histogram for angle 5 sample image of noon time frame. These Figures depict a random sample of images of their RGB distributions.

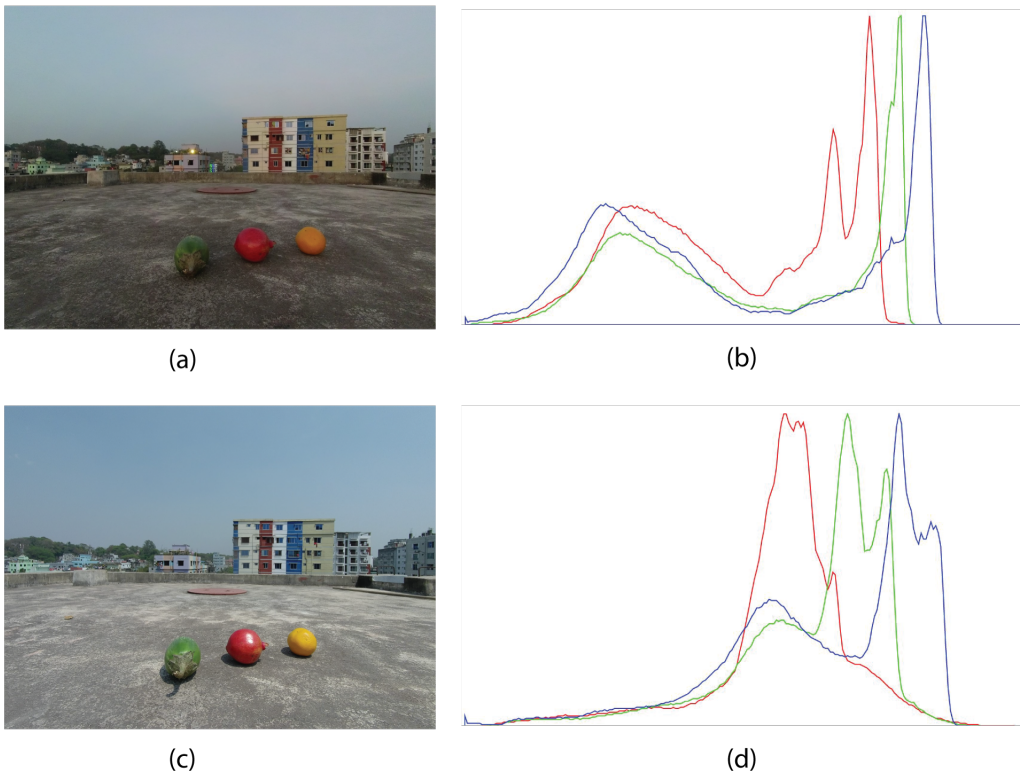


Figure 3.8: Image and histogram representation for 1920 x 1080 pixel size image for two different time frame (a) original sample image of afternoon time frame (b) histogram representation of image in first section of afternoon time frame (c) original sample image of noon time frame (d) histogram representation of image in third section of noon time frame.

3.7.3 RGB vs HSV Color Space

Following Figure 3.9 represents a 3D scatter plot for RGB and HSV values. HSV value represents a different color space than RGB. It consists of Hue, saturation and value. These diagrams are represented to get an in depth idea about our dataset. These are sample images randomly taken from our dataset and a random selection of angles. To avoid redundancy of visualization the model has opted to only provide a sample 3D scatter plot.

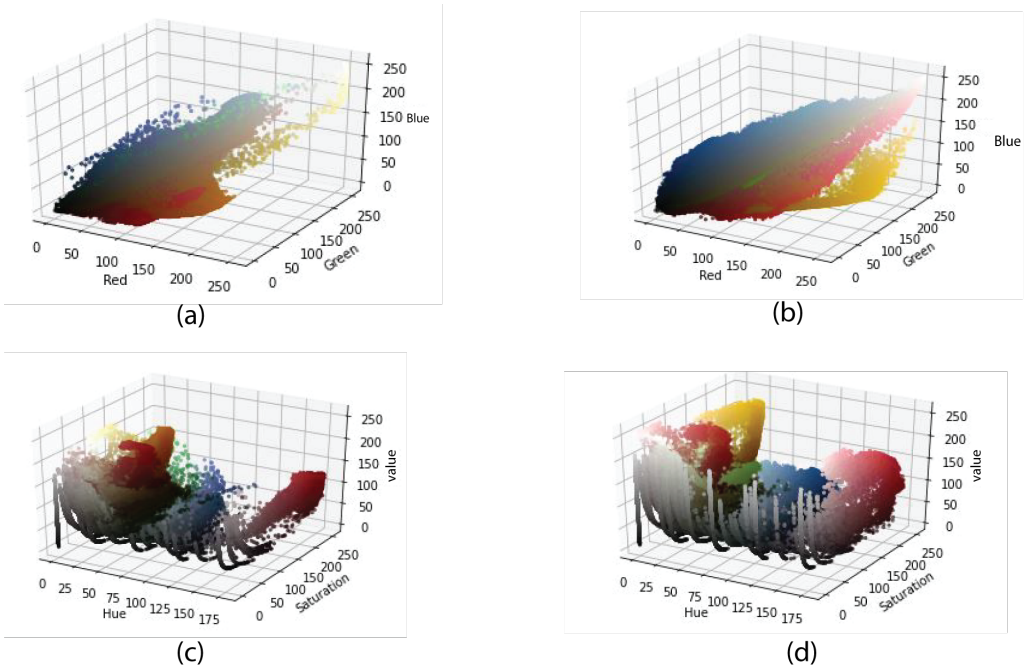


Figure 3.9: RGB vs HSV Color Space representation in 3D scatter plot for original image in 1920 x 1080 pixels in Figure 3.8 (a) RGB scatter plot for original image of the first section of Figure 3.8 (b) HSV scatter plot for original image of the first section of Figure 3.8 (c) RGB scatter plot for original image of the third section of Figure 3.8 (d) HSV scatter plot for original image of the third section of Figure 3.8.

3.7.4 BGR to RGB Conversion

Opencv platform is used in this model. Also, for importing images to minimize any import loss of image. However, Opencv loads in any given image in BGR (Blue-Green-Red) format. Therefore, This model has used Opencv's built-in one of the 150 color space conversion methods. Here, BGR2RGB method is utilized to import images directly to RGB format without losing my value. In this part of data pre-processing, we have converted the BGR image to RGB as OpenCv loads images in default BGR format. It does not alter the image format nor the RGB distribution. It has converted all of the dataset into RGB format before loading into this model. Figure 3.10 depicts the transformation of an image from BGR color space to RGB color space for a sample image of noon time frame setting.

Figure 3.10: an original image in the noon from two different color space (a) BGR color space of the original image (b) RGB color space of the original image

3.7.5 Image Resize

We have created our dataset in Full HD format which has a pixel value of 2073600 (1920x1080). However, due to hardware limitations we had to resize our image into a smaller image size. Moreover, we have used two different pixel values that are 148x196 (29008) pixels which is 380x420 (159600) pixels to show that with higher pixel value better results can be achieved. With two different pixel ratios, different conditions and outcomes of our model can be determined through plotting. We have trained our dataset for two epochs for each pixel ratio so that we can determine the variation in our training environment.

Benefits of Using Autoencoder in the projected problem scenario:

Data-specific: An AE is only capable of compressing data similar to what they are trained on. An AE is not similar to a standard data compression algorithm like gzip as it learns specific features for a given specific set of data. An AE trained on handwritten digits can not compress it to landscape photos.

Reconstruction Loss: It is a method that measures how well an AE is performing and difference between the input and output.Then the model attempts to minimize

the reconstruction loss using backpropagation method.

Unsupervised: Training an AE is not that much complex. As it is an unsupervised learning technique this is why an AE doesn't require explicit labels to train on. An AE creates its own label to get trained on a specific set of data. It can be called self-supervised learning.

An AE is primarily a fully connected layer, through this layer input construct the bottleneck layer or the code. This is mainly the encoder. The decoder part is also an ANN architecture which constructs the output from the bottleneck layer. The output should be identical to the input. Actually encoder and decoder are the mirror image of each other, but this is not a must. The only compulsory requirement of an AE is to have the dimensionality of the input and result must be similar . The middle part can be fixed with trial and test basis.

Here comes the hyperparameter part of the AE. Hyperparameters are the variables which signifies the network architecture and the variables which tells how the network is trained. In this case, the network structure is the number of layers or number of nodes per layer and the variables are loss function, learning rate, number of epochs etc. Primarily, There are four hyperparameters that is needed to define before training an AE.

Code size: Quantity of nodes in the mid point layer or the bottleneck layer. Smaller size results in more condensation.

Quantity of nodes for each layer: The quantity of nodes per layer reduces with every subsequent layer of the encoder, and heightens back in the decoder. Also the decoder is uniform to the encoder in the structure of layers. As determined above this is vital and we have total dominance over the parameters. Using a number of hidden units lower than inputs forces AE to learn a compressed approximation [44].

Loss function: There are few loss functions in the neural network. Such as mean squared error (mse), binary cross entropy. If the input values are in the range between 0 and 1, the cross entropy can be used, otherwise the mean squared error is used.

Chapter 4

Results and Discussion

The output of this model is not resized to 1920 x 1080 pixels due to the fact that the model mainly works on the resized image of 380 x 420 pixels and 148 x 196 pixels as input and runs the autoencoder model to eventually train and validation test in resized image size. Hence, fully scaling the output image to original image size which is 1920 x 1080 would stretch and make false RGB values. RGB value would further deteriorate as the model is making a huge amount of scaling back to the original image and we would get a distorted image in output. Therefore, due to hardware limitations of eventually training and testing 1920 x 1080 pixels the method has to work with smaller image sizes of 380 x 420 and 148 x 196 pixels to train and test this model. In Figure 4.2 the result for 380 x 420 pixel represents the output due to the fact that this model works better for higher pixel ratio. However, the proposed methodology has also trained and tested this model for a smaller pixel 148 x 196 to represent the fact and compare that this model works better for larger pixel proportions which is represented in Figure 4.3 for smaller pixel. A particular aspect ratio of 148 x 196 and 380 x 420 pixel is chosen for the convenience of this model to produce better output as pixel sizes maintain a distinctive pattern of RGB values. This model has verified and defined different aspects through cardinal directions which includes in Figure 4.1 the directions of north, south, east and west. The model is completed to objectify the difference of lighting conditions in various directions due to the sun.

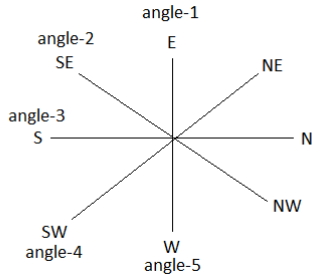


Figure 4.1: Cardinal directions defining our angles

a. Original Image: This section represents the raw data that were originally obtained as part of our dataset in 1920 x 1080 pixel ratio. Section (a) of Figure 4.2 and Fig 4.3 is segmented in five different angles which are labeled as angle 1, angle 2, angle 3, angle 4 and angle 5. These images are clicked in an afternoon environment

which according to Bangladesh time zone represents the reduction of overall lighting condition and eventually leading to sunset (dark lighting condition as dark is also a color spectrum in RGB color space). According to the section (a) we can see that angle 1 has the most vibrant color setting whereas angle 5 has the least color vibrancy.

b. Resized input image: section (b) in Figure 4.2 and Figure 4.3 represents the resized image ratio before being loaded into the AE model. This section represents the reduction in pixel size to 380 x 420 and 148 x 196 pixels for Figure 4.1 and Figure 4.2 in the afternoon in contrast to the original image. The section also contains 5 angles as the previous section (a) and these images are fed into the first encoding layer of our architecture for the training to start. Here, we have chosen afternoon images as input to show the effectiveness of our model as according to general color space problems it is easier to reduce the lighting condition of an image. Whereas it is difficult to increase the lighting condition of an image to make it as close to the original image of an noon setting. Therefore, we have chosen to show the output of noon setting of images with given input of afternoon images.

c. Output image: This section contains the images that are output images. In Figure 4.2 and Figure 4.3 the output of the images are differentiated by 5 angles. The results we get from training our model are in the Noon setting here. Noon setting represents a more vibrant lighting condition which is basically before 12 pm according to Bangladesh timezone. Here, we can differentiate the difference between afternoon angle 3 image in section (a) which is input image and noon angle 3 image in section (b) which is the output image. Lighting conditions can also be noticed under different angles of section (a) and section (b). The output here is represented in 380x420 for Figure 4.2 and 148 x 196 pixel size Figure 4.3.

4.1 Experimental Results for two different image dimensions with five different angular images as test inputs

The model has tested two different image spaces with five angular images as test inputs. The experimental result shows that images with resolution provide reconstructed image output with higher quality. Figure 4.1 shows original input images, resized input images and reconstructed output images using the proposed model with the resolution of 380 x 420 pixel whereas captured images are with resolution of 1920 x 1080 pixel. Both for training and testing we used resized images as input to autoencoder technique.

Similarly, we tested the proposed model using the images with lower resolution, 148 x 196 pixel both for training and testing. Figure 4.2 and Figure 4.3 presents the results of this experiment for both pixel 380x420 and pixel 148x196.

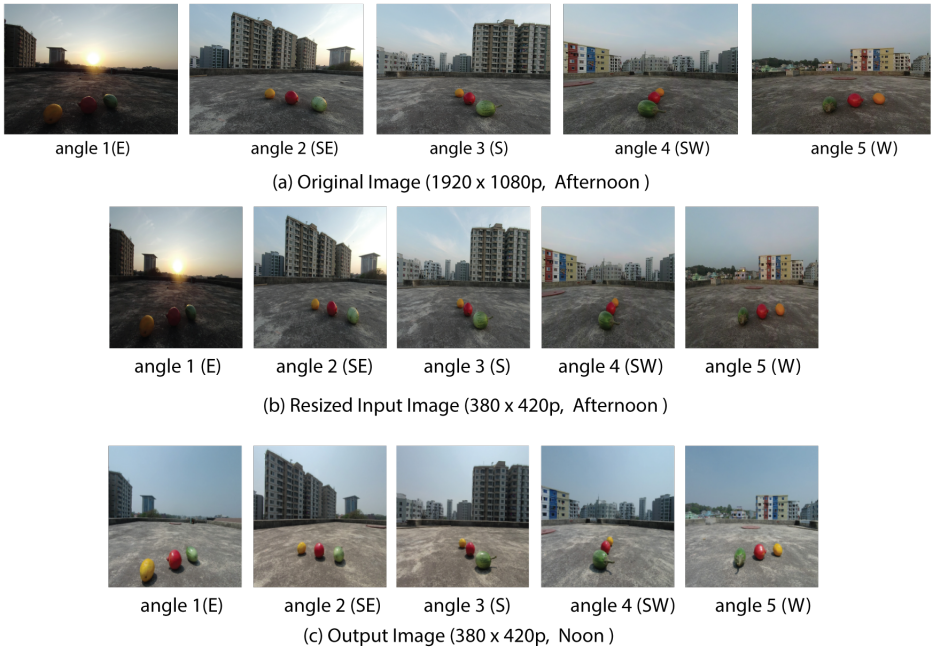


Figure 4.2: Sample results of proposed model for 380 x 420 pixel (a) original test input images from five different angles (b) resized input images from the original images (c) reconstructed output images from afternoon images to noon images.

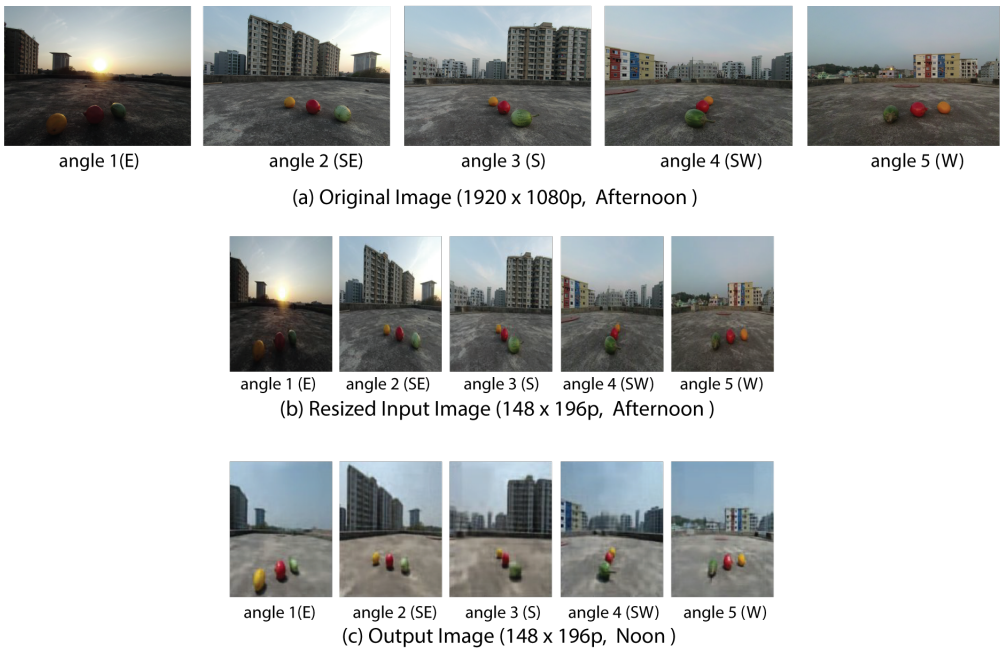


Figure 4.3: Sample results of proposed model for 148 x196 pixel-(a) original test input images from five different angles (b) resized input images from the original images (c) reconstructed output images from afternoon images to noon images.

From Figure 4.2 and Figure 4.3 this can be observed that the more resolution input dataset provide the higher quality reconstructed output images can be obtained. Moreover, the histogram and other analysis is presented in the Result Analysis subsection.

4.2 Result Analysis

Images and figures represent the analysis and visualization of our model. The analysis of our results provide validity to the proposed.

4.2.1 MSE Representation for Lower Epoch

MSE represents mean square error during the training and validation testing period. It shows the squared estimation between the actual and estimated value. Here, the model has shown MSE representation for lower epoch which is 20. The following graphs (Figure 4.4. and Figure 4.5) show mean square error for angle 1 and angle 5. Angle 1 and angle 5 represent east and west direction respectively.

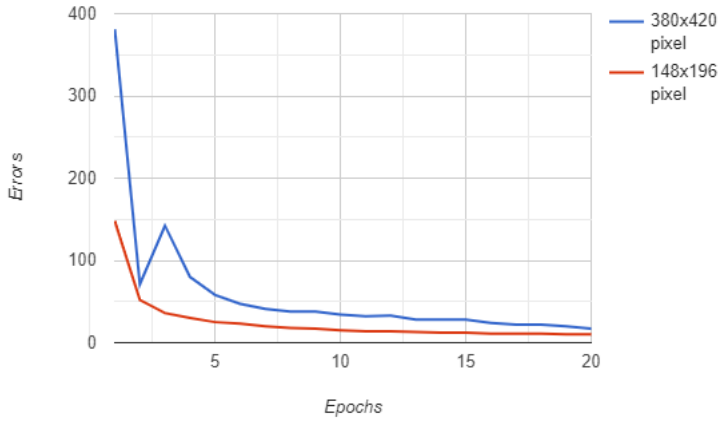


Figure 4.4: MSE vs Epoch graph for angle1(E) from afternoon to noon.

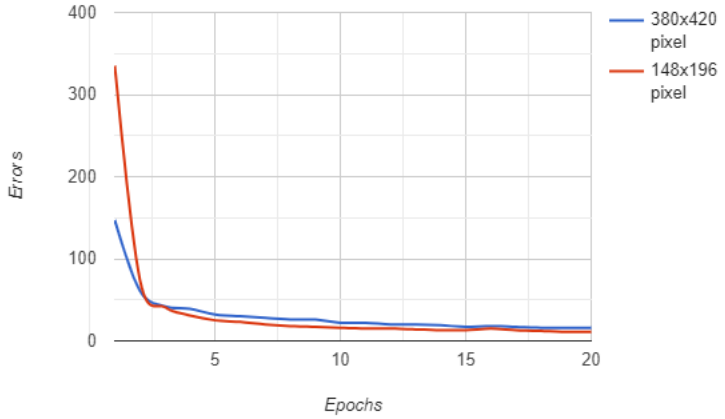


Figure 4.5: MSE vs Epoch graph for angle1(E) from noon to afternoon.

This epoch graph determines the training difference between different pixel sizes of 380 x 420 and 148 x 196. In both the above figures it has shown representation for 20 epochs and its corresponding mean square error. Here , this can be seen that with increased epoch our model's error is decreasing. The curve almost flattens and remains the same after the fifth epoch in terms of Figure 4.5 in the noon to afternoon training whereas there is a sudden increase and decrease of MSE in Figure 4.4 for pixel value 380 x 420.

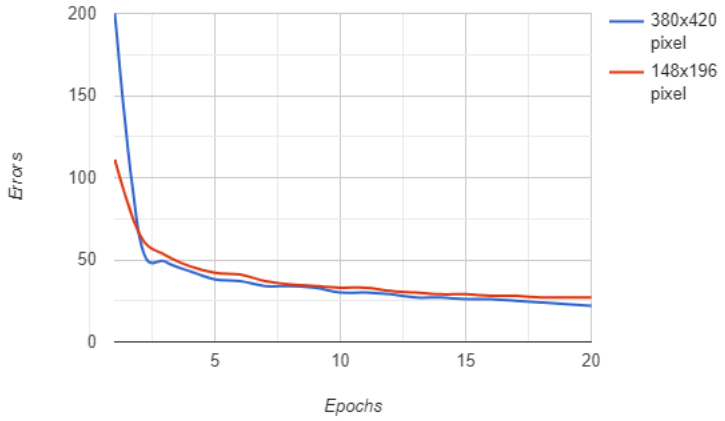


Figure 4.6: MSE vs Epoch graph for angle5(W) from afternoon to noon.

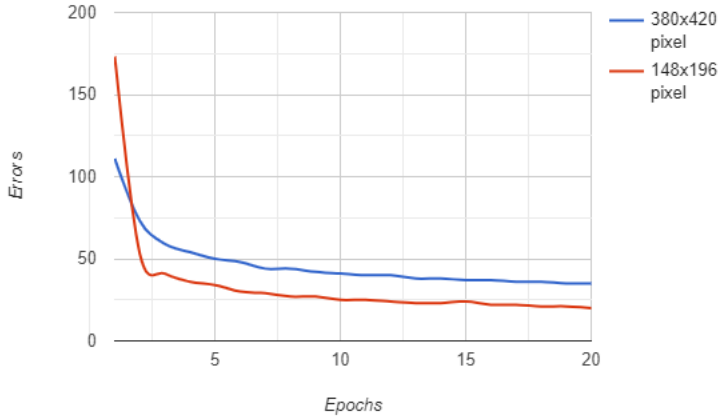


Figure 4.7: MSE vs Epoch graph for angle5(W) from noon to afternoon.

Figure 4.6 and Figure 4.7 shows MSE(mean square error) vs epochs graph is shown for the training domain of Afternoon to Noon(angle 5) and Noon to Afternoon(angle 5) respectively. Here, angle 5(W) represents the west direction. This epoch graph determines the training difference between different pixel sizes of 380 x 420 and 148 x 196. In both the above figures we have shown representation for 20 epochs and its corresponding mean square error. Here , it can be seen that with increased epoch this model's error is decreasing. The curve almost flattens and remains the same after the seventh epoch. In Figure 4.7 MSE is higher for 380 x 420 pixels rather than 148 x 196 due to the higher amount of RGB processing per pixel domain.

4.2.2 MSE Representation for Higher Epoch

MSE represents mean square error during the training and validation testing period. Here, it has shown MSE representation for higher epoch which is 40. The following graphs show mean square error for angle 1(E) and angle 5(W).

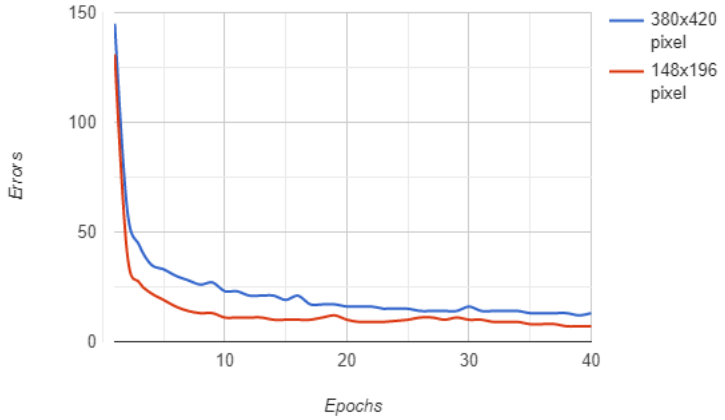


Figure 4.8: MSE vs Epoch graph for angle1(E) from afternoon to noon.

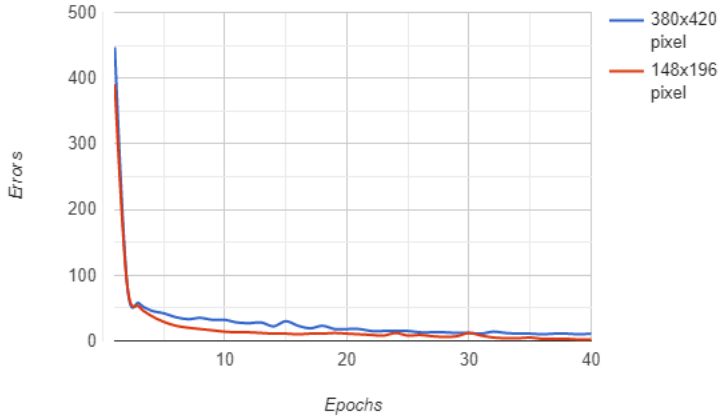


Figure 4.9: MSE vs Epoch graph for angle1(E) from noon to afternoon.

Figure 4.8 and Figure 4.9 MSE(mean square error) vs epochs graph is shown for the training domain of Afternoon to Noon(angle 1) and Noon to Afternoon(angle 1) respectively. This epoch graph determines the training difference between different pixel sizes of 380 x 420 and 148 x 196. In both the above figures it is a representation for 40 epochs and its corresponding mean square error. Here , this is shown that with increased epoch our model's error is decreasing over the epoch iterations and almost flattens for Figure 4.9. In Figure 4.5 380 x 420 pixel has a higher MSE than Figure 4.9 due to higher image values needed to process. Moreover, it also determines that due to difficulty to generate lighting conditions similar to that of afternoon more computation is needed where output image is of noon time frame with a higher light intensity Figure 4.8.

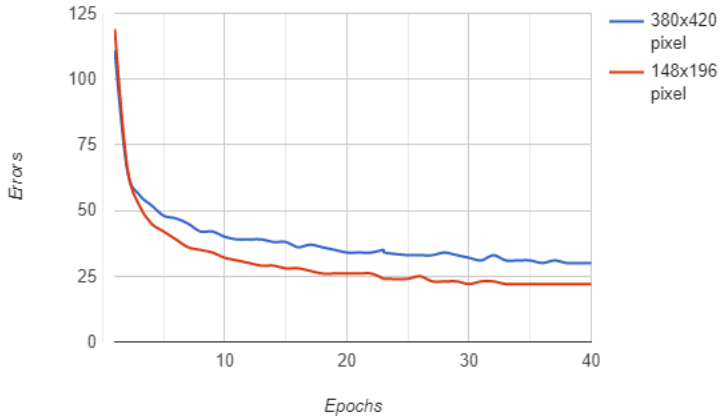


Figure 4.10: MSE vs Epoch graph for angle5(W) from afternoon to noon.

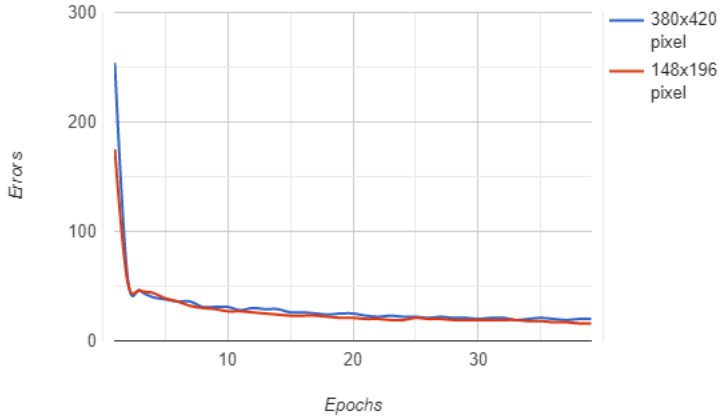


Figure 4.11: MSE vs Epoch graph for angle5(W) from noon to afternoon.

On the other hand, In above Figure 4.10 and Figure 4.11 MSE (mean square error) vs epochs graph is shown for the training domain of Afternoon to Noon (angle 1) and Noon to Afternoon (angle 1) respectively. This epoch graph determines the training difference between different pixel sizes of 380 x 420 and 148 x 196. In both the above figures we have shown representation for 40 epochs and its corresting mean square error. Here, it can be observed that with increased epoch our model's error is decreasing over the epoch iterations and almost flattens for Figure 4.11. In Figure 4.10, 380 x 420 pixel has a higher MSE than Figure 4.11 due to higher image values needed to process. The later figure is more linear due to the fact that more information is needed to process when converting an image to higher intensity lighting condition values. Therefore, the higher pixel value has also the higher amount of MSE corresponding to the epoch variation in Figure 4.10.

4.2.3 RGB Histogram

RGB histogram is a way of representing an image in pure red, green and blue color space. This methodology has shown the RGB histogram for 380 x 420 pixels and 148 x 196 pixels in Figure 4.12 and Figure 4.13 respectively. These histograms represent the color reproduction capacity of the proposed model. Here, angle 3 represents the south direction.

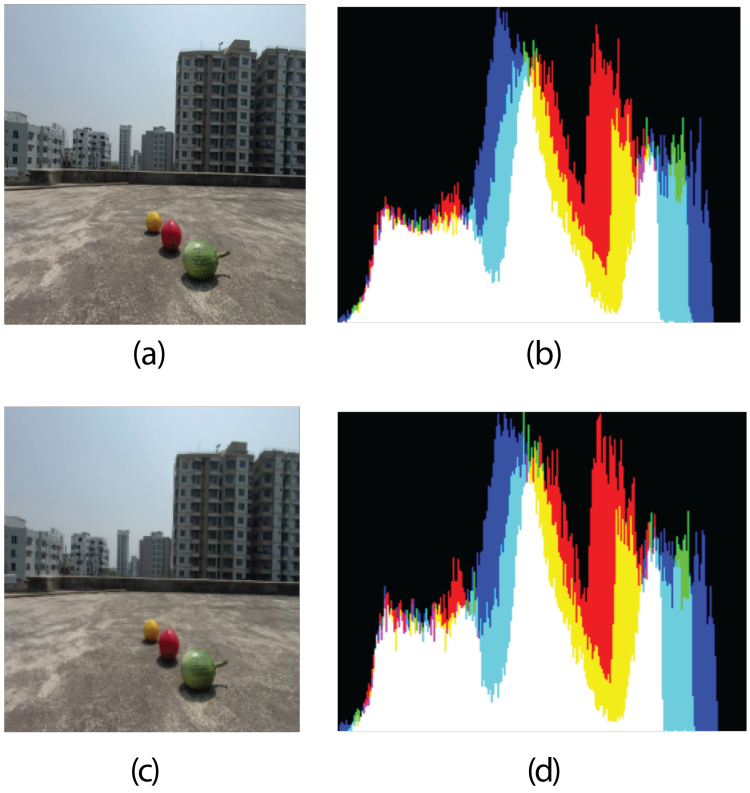


Figure 4.12: RGB histogram for 380 x 420 pixel original and output image noon time frame for angle 3(S) direction (a) Original resized image of 380 x 420 pixel (b) Histogram of original resized image (c) Output image in 380 x 420 pixel on given input of afternoon angle 3(S) direction (d) Histogram of resized output image in 380 x 420 pixel.

Figure 4.12 section (a) depicts the original resized image size of 380 x 420 and its RGB histogram in the right image in section (b) for angle 3 in noon lighting condition. In section (c) depicts the output image size of 380 x 420 pixel and its RGB histogram on the right for the image in noon lighting condition for section (d). This representation helps us understand that after giving the input image of afternoon for angle 3(W) the output is quite similar to the original image of noon angle 3. Here, the original image that was used to train our model is the image represented in section (a). On completion of training of our model after giving an input image of afternoon time frame setting for angle 3(S) this research gets an output for noon angle 3 time frame lighting condition and similar features for RGB values. RGB histograms help us utilize the condensation of values to determine the visual accuracy of our model. Between the histogram of section (b) and section (d) that can be seen slight change in spikes of RGB values which show that after

giving input image for afternoon time frame setting (angle 3) which get an output image of noon time frame condition(angle 3) which has similar characteristic with the original noon time frame condition and setting (angle 3).

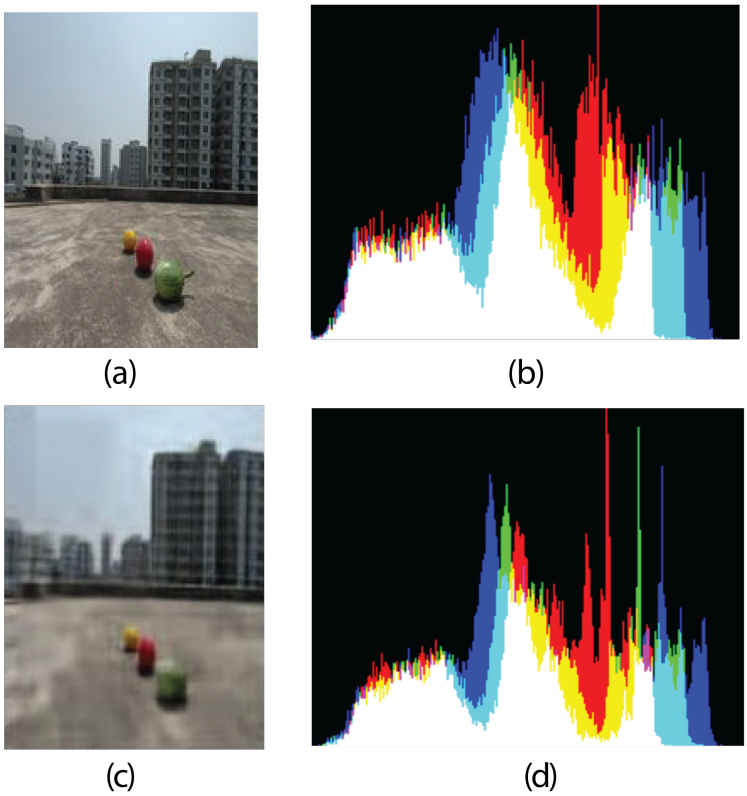


Figure 4.13: RGB histogram for 148 x 196 pixel original and output image noon time frame for angle 3(S) direction (a) Original resized image of 148 x 196 pixel (b) Histogram of original resized image (c) Output image in 148 x 196 pixel on given input of afternoon angle 3(S) direction (d) Histogram of resized output image in 148 x 196 pixel

Figure 4.13, section (a) depicts the original resized image size of 148 x 196 pixel and its RGB histogram in the right image in section (b) for angle 3 in noon lighting condition. In section (c) depicts the output image size of 148 x 196 and its RGB histogram on the right for the image in noon lighting condition for section (d). This representation helps to understand that after giving the input image of afternoon angle 3(S) the output which has RGB edges to the original image of noon angle 3(S). Here, the original image that is used to train our model is the image represented in section (a). On completion of training of our model after giving an input image of afternoon time frame setting for angle 3 we get an output for noon angle 3 time frame lighting condition and we get different features for RGB values. RGB histograms help us utilize the condensation of values to determine the visual accuracy of our model. Between the histogram of section (b) and section (d) which indicates large change in spikes of RGB values which show that after giving input image for afternoon time frame setting (angle 3) the output image of noon time frame condition(angle 3) which has similar characteristic with the original noon time frame condition and setting (angle 3).

4.2.4 Supersampled Heatmap

Supersampled heatmap here represents the color depth and effect of sunlight across different objects. It also can be used to determine the effect and variation of colors in a color image. Figure 4.14 and Figure 4.15 shows supersampled heatmap analysis for 380 x 420 and 148 x 196 pixels respectively.

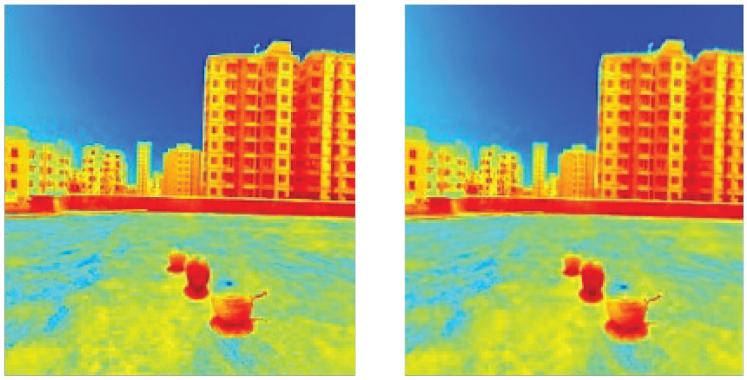


Figure 4.14: Supersampled Heatmap for 380 x 420 pixel (a) original resized image of noon time frame setting in angle 3(S) direction (b)output image of noon time frame in angle 3(S) direction on given input of afternoon time frame setting of angle 3(S) image.

Figure 4.14 shows two different heatmaps of supersampled images for noon angle 3(S). The first figure depicts the original image and the second figure depicts the output image for 380x420 pixels. Heatmap images help determine the exposure difference in lighting conditions. Here, the first figure represents the exposure of objects and color spaces under exposure of noon lighting conditions for the original image. And the later figure defines the exposure of color spaces of noon angle 3(S) timeframe lighting conditions if the input image is of afternoon angle 3(S) time frame lighting condition.The supersampled heatmap version here depicts descent visual similarity with the original image.

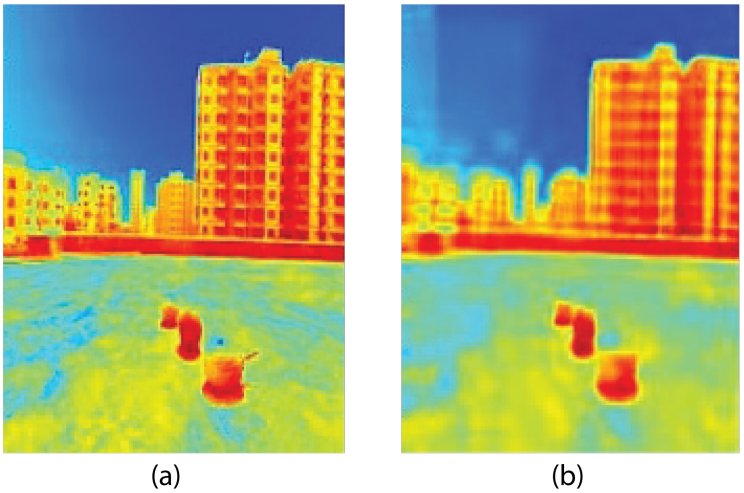


Figure 4.15: Supersampled Heatmap for 148 x 196 pixel (a) original resized image of noon time frame setting in angle 3(S) direction (b)output image of noon time frame in angle 3(S) direction on given input of afternoon time frame setting of angle 3(S) image.

In Figure 4.15 it has provided two different heatmaps of supersampled images for noon angle 3. The first figure depicts the original image and the second figure depicts the output image for 148 x 196 pixel. Heatmap images help determine

the exposure difference in lighting conditions. Here, the first figure represents the exposure of objects and color spaces under exposure of noon lighting conditions for the original image and the later figure defines the exposure of color spaces of noon angle 3 timeframe lighting conditions if the input image is of afternoon angle 3 time frame lighting condition. The supersampled image on the right is distorted than the original image due to the fact that it has lesser pixel value than the original image.

4.2.5 RGB Channel

RGB channel segmentation helps understand the depth and variation of effect of red, green and blue color domain in a RGB color space. Following Figure 4.16 and Figure 4.17 show the representation for RGB channel segmentation for 380 x 420 pixels and 148 x 196 pixels.

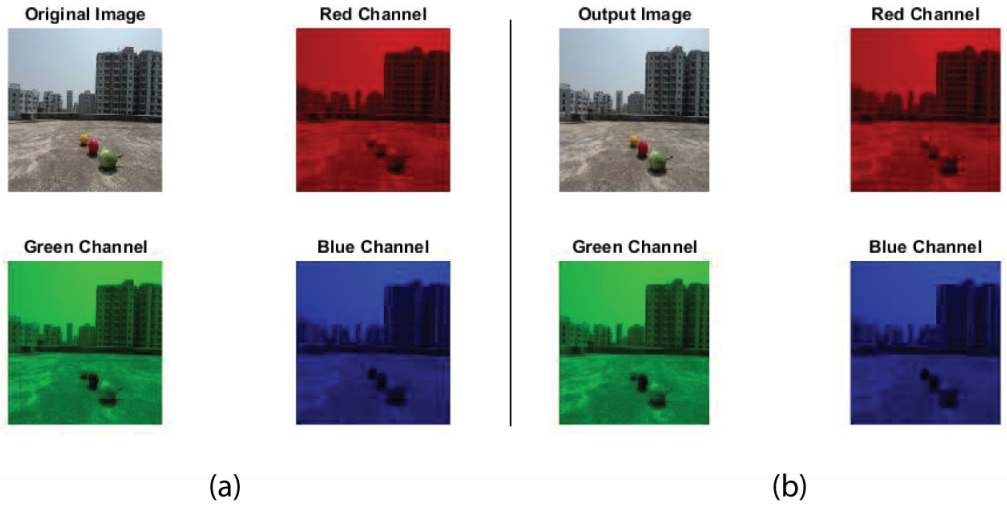


Figure 4.16: Red, Green and Blue channel segmentation for 380 x 420 pixels (a) channel segmentation of original resized image of noon time frame setting in angle 3(S) direction (b) channel segmentation of output image of noon time frame in angle 3(S) direction on given input of afternoon time frame setting of angle 3(S) image.

Figure 4.16 shows separate red, green and blue channel segments for afternoon angle 3 original and output image of size 380x420. Output image is obtained through taking the after noon angle 3 time frame setting image as an input. These channels determine the variation that can be observed visually to determine channel variations of both original and output images. Both figures determine the red, green and blue segmentation of the original image in 380 x 420 size. Here, the segmentation of channels helps us to visualize the impact of the RGB channels on the main image. The channels verify the intensity of their respective values on the overall image. Both of the figures have significant effect on the lighter green channel and a darker blue channel which explains the effect of blue channel image on the whole image.

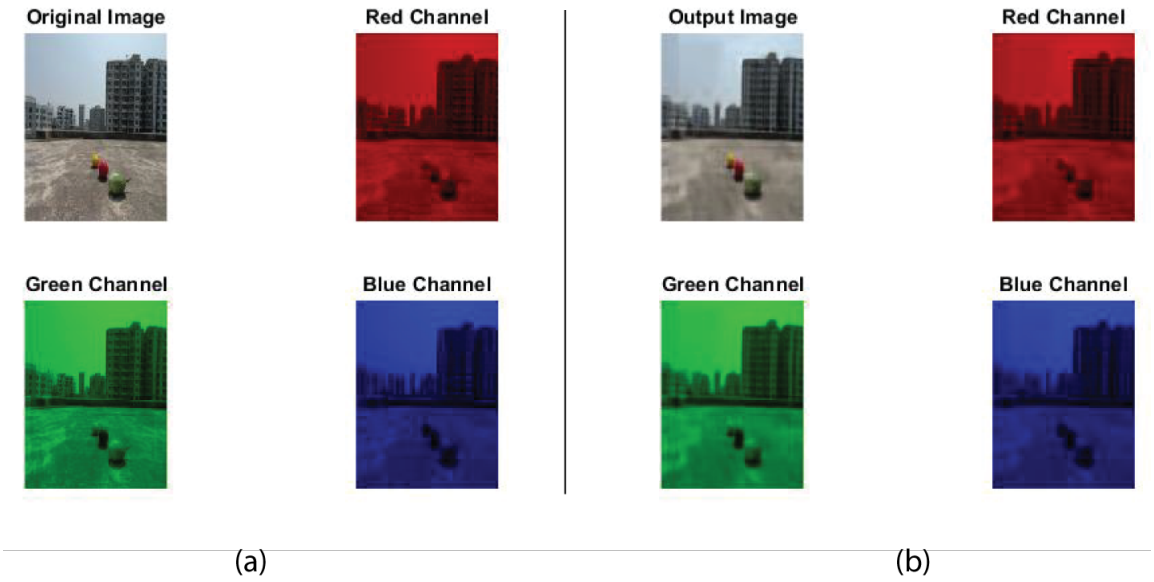


Figure 4.17: Red, Green and Blue channel segmentation for 148 x 196 pixels (a) channel segmentation of original resized image of noon time frame setting in angle 3(S) direction (b) channel segmentation of output image of noon time frame in angle 3(S) direction on given input of afternoon time frame setting of angle 3(S) image.

In Figure 4.17 this model created separate red, green and blue channel segments for noon angle 3 original and output image of size 148 x 196. Output image is obtained through taking the noon angle 3 time frame setting image as an input. These channels determine the variation that can be observed visually to determine channel variations of both original and output images. Both figures determine the red, green and blue segmentation of the original image in 148 x 196 size. The channels verify the intensity of their respective values on the overall image. Both of the figures have significant effect on the red channels and a less darker blue channel than noon 3 time frame setting.

4.2.6 RGB Matrix Evaluation

Figure 4.18 shows the sampling of RGB value matrix is shown to differentiate between the original resized image and output image of the noon time frame setting of angle 3(S) for 380 x 420 pixels. Only representation of 380 x 420 pixels is given as according to the above analysis matrix evaluation is a way of further proving the fact of better output in 380 x 420 pixels.

In section (c), a matrix is shown for the red object in the image. The sampled approximate pixel value for red range from 110 to 154, green range from 13 to 36 and blue ranges from 27 to 54. The value for images in section (e) for the output image of noon angle 3 for the given input image of afternoon angle 3 time frame setting we get the RGB matrix value for red object. The approximate value for red ranges from 93 to 117, green from 13 to 16 and red from 22 to 29.

In section (d), a matrix is shown for the green object in the image. The sampled approximate pixel value for red range from 54 to 70, green range from 75 to 87 and

blue ranges from 44 to 53. The value for images in section (f) for the output image of noon angle 3 for the given input image of afternoon angle 3 time frame setting we get the RGB matrix value for green object. The approximate value for red ranges from 39 to 52, green from 57 to 66 and red from 30 to 46.

From the above comparison it can be evaluated the fact that the proposed model produces better output in terms of rgb values in the respective problem domain. The values are close and form almost similar dimension of images in terms of RGB color reconstruction where the values do not deviate much from the original value.

The sunlight condition is also adjusted through an unsupervised learning model to adjust the values to accurate lighting conditions.

(a)

R:136 G: 21 B: 38	R:132 G: 19 B: 37	R:137 G: 21 B: 42	R:147 G: 29 B: 47	R:150 G: 32 B: 49	R:154 G: 36 B: 54
R:124 G: 18 B: 34	R:120 G: 16 B: 32	R:119 G: 12 B: 30	R:129 G: 17 B: 33	R:133 G: 20 B: 35	R:134 G: 23 B: 39
R:110 G: 13 B: 27	R:109 G: 12 B: 27	R:108 G: 10 B: 26	R:110 G: 11 B: 26	R:116 G: 14 B: 27	R:117 G: 14 B: 29

(c) Sample RGB matrix for red coloured object

R: 74 G: 90 B: 53	R: 58 G: 74 B: 36	R: 61 G: 77 B: 39	R: 71 G: 88 B: 50	R: 68 G: 86 B: 49	R: 59 G: 75 B: 44
R: 60 G: 74 B: 39	R: 59 G: 73 B: 36	R: 67 G: 82 B: 47	R: 66 G: 82 B: 47	R: 68 G: 84 B: 46	R: 49 G: 64 B: 35
R: 74 G: 87 B: 44	R: 49 G: 62 B: 24	R: 40 G: 54 B: 21	R: 40 G: 54 B: 23	R: 42 G: 55 B: 21	R: 34 G: 47 B: 17

(d) Sample RGB matrix for green coloured object

(b)

R:108 G: 12 B: 26	R:105 G: 10 B: 26	R:105 G: 11 B: 25	R:110 G: 13 B: 27	R:113 G: 13 B: 28	R:117 G: 16 B: 29
R: 97 G: 12 B: 23	R: 95 G: 11 B: 23	R: 94 G: 10 B: 22	R: 98 G: 11 B: 23	R:101 G: 11 B: 24	R:103 G: 13 B: 24
R: 93 G: 13 B: 22	R: 90 G: 10 B: 21	R: 88 G: 9 B: 21	R: 90 G: 9 B: 22	R: 92 G: 10 B: 23	R: 93 G: 11 B: 21

(e) Sample RGB matrix for red coloured object

R: 48 G: 64 B: 32	R: 52 G: 70 B: 38	R: 55 G: 71 B: 39	R: 52 G: 68 B: 38	R: 49 G: 66 B: 39	R: 39 G: 57 B: 30
R: 59 G: 75 B: 37	R: 67 G: 84 B: 46	R: 66 G: 81 B: 42	R: 51 G: 68 B: 35	R: 46 G: 61 B: 34	R: 46 G: 62 B: 34
R: 52 G: 66 B: 30	R: 53 G: 66 B: 32	R: 56 G: 68 B: 35	R: 42 G: 56 B: 28	R: 40 G: 55 B: 27	R: 37 G: 51 B: 22

(f) Sample RGB matrix for green coloured object

Figure 4.18: RGB matrix evaluation for noon time frame image in angle 3 (S) direction of 380x420 pixels (a) original resized image (b) Output image of noon time frame in angle 3(S) direction on given input of afternoon time frame setting of angle 3(S) direction image (c) sample rgb matrix for red coloured object for original resized image (d) sample rgb matrix for green coloured object for original resized image. (e) sample rgb matrix for red coloured object for output image. (f) sample rgb matrix for red coloured object for output image.

4.2.7 3D Scatter Plot Representation

As our research works on color vision approach we have mainly focused on RGB color space. However, this research paper also tried to show the difference of output in 380 x 420 pixels and 148 x 196 pixels for both RGB and HSV color space in 3D scatter plot format. 3D scatter plot can help us determine in terms of actual value distribution in color spaces. Figure 4.19 and Figure 4.20 represent the 3D scatter plot for both color spaces in two different color spaces.

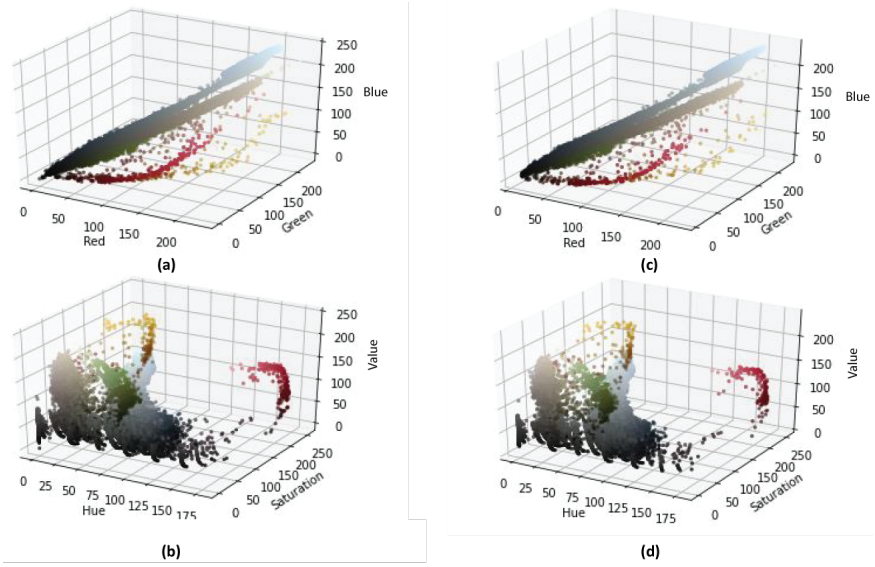


Figure 4.19: 3D scatter plot diagram for 380 x 420 pixel original and output image for noon angel 3(S) time frame (a) RGB scatter plot of original resized image (b) HSV scatter plot of original resized image of angle 3 (S) direction image (c) RGB scatter plot of output image on given input of afternoon time frame setting of angle 3(d) HSV scatter plot of output image on given input of afternoon time frame setting of angle 3.

Figure 4.19 depicts the RGB and HSV scatter plot representation of our input and output image. Here, it can be seen the RGB value does not defer much in terms of original and output image as the distribution is almost similar. Therefore, the proposed model works well for 380 x 420 pixels. For HSV scatter plot distribution the changes are also very low as the hue, saturation and value depict almost similar in terms of original resized image and output image on given input of afternoon angle 3(S).

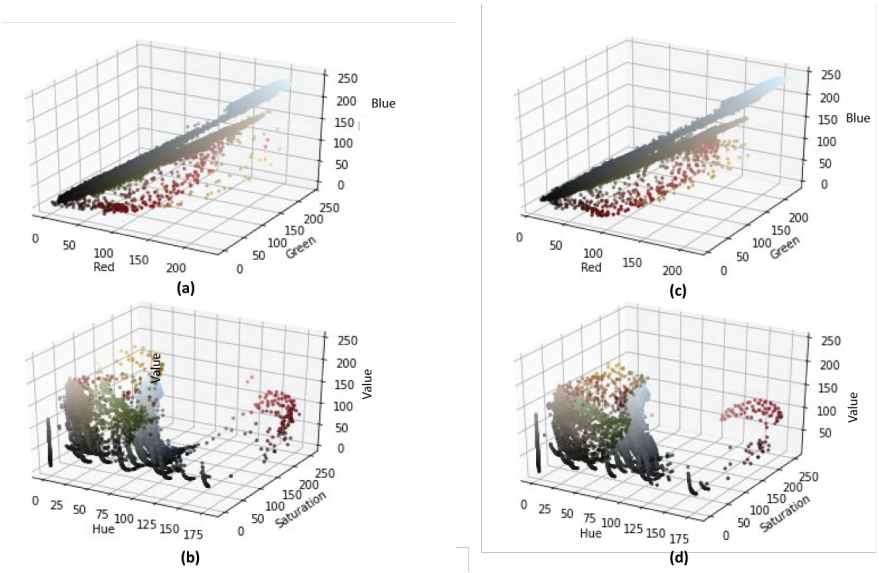


Figure 4.20: 3D scatter plot diagram for 148 x 196 pixel original and output image (a) RGB scatter plot of original resized image (b) HSV scatter plot of original resized image(S) direction image (c) RGB scatter plot of output image on given input of afternoon time frame setting of angle 3 (d) HSV scatter plot of output image on given input of afternoon time frame setting of angle 3.

Figure 4.20 depicts the RGB and HSV scatter plot representation of our input and output image. Here, it can observe that the RGB value defer much in terms of original and output image as the distribution is different in some areas in terms of density in scatter plot. The proposed model provides an average result for 148 x 196 pixels. For HSV scatter plot distribution the changes are also high as the hue, saturation and value depict differently in terms of original resized image and output image on given input of afternoon angle 3(S).

From the RGB histogram that we have shown we can know the difference between the input and output image that we have trained. We have used supersampled heatmap images to determine the effect of variation of exposure over various locations in the images. RGB channeling and matrix evaluation we used to determine the effects on training on our output image.

After training our autoencoder based model and training our dataset of 10,000 images we have only selected mainly showed the visualisation of input data and output data through some selected angle and timeframe to better grasp the changes and outcomes of our model.

One of the drawbacks of our procedures include the fact of not being able to properly indicate the accuracy rate through regular accuracy algorithms or methodology. Our research being visual centered makes it difficult for our dataset and outcomes to measure through the normal standards. Therefore, we have shown various orientations and various image visualizations to further strengthen our claim and support the model that we have created.

Despite having certain limitations our model proves better training and testing of images in case of multi angle training and model verification. The input image of a given angle of a time frame is generally brought to the output of the image in another time frame.

From the gathered visualizations we can determine that our model provides good accuracy to the given problem domain and works well for a multi angle framework as we are working on five different angles facing in different directions.

It can also be confirmed from visualizations that our model provides better results for higher resolution images as we get a better result for 380 x 420 pixels than the smaller 148 x 196 pixels. Our different lighting condition is also objectified by the RGB values of the output images which are trained by our model.

Chapter 5

Conclusion

This research mainly focuses on reconstructing images to normalize into a predefined form under various conditions of color vision or color machine vision. The main challenge often occurs while analyzing an image from a low light state is to imply different techniques when the outcome is not up to the mark in many cases. As the reflection of light is very tiny from the image. Thus, detection of an object from an image needs more affordance. The purpose of this paper is to overcome the challenge in an efficient way. In this paper one unique method is introduced. By this approach any image which has a lack of light can turn into a brighter one as the image is taken in the daylight. That is, the image does not need further implication of different methods or algorithms which makes detecting an object easier. In this approach, an image is taken as input of SAE which feeds the encoding part. The image goes through several hidden layers of encoding parts and reaches its bottleneck after some features extraction. The bottleneck then appears as the input of the decoding part of SAE. Similarly, it goes through several layers of decoding parts and is ready for showing the output. The significant observation is the output is close to its input after the implication of this method. However, several techniques are used to show the output is close enough to its input. This paper shows the amount of loss using graph for two different pixel images, one is 148x196 pixels and the other one is 340x420 pixels. The graph indicates that the amount of loss of input to output is decreasing after some epochs. It is calculated for two different amounts of epochs 20 and 40. Also, the histogram we use to show the difference of input and output is almost similar as the model works perfectly. Likewise, supersampled heatmap, RGB color variation, 3D scatter plot and RGB color code are covered to show the accuracy of the output against its input and original image. In general, the paper proposes a new approach on reconstructing an image which yields promising results.

Reference

[1] K. Jhang and J. Cho, "Cnn training for face photo based gender and age group prediction with camera", in *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2019, pp. 548–551.

[2] I. Benkhaled, I. Marc, and D. Lafon, "Colour contrast detection in chromaticity diagram: A new computerized colour vision test", in *2017 IEEE Western New York Image and Signal Processing Workshop (WNYISPW)*, 2017, pp. 1–4.

[3] T. Nishi, S. Kurogi, and K. Matsuo, "Grading fruits and vegetables using rgb-d images and convolutional neural network", in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017, pp. 1–6.

[4] B. Öztürk, M. Kirci, and E. O. Güneş, "Detection of green and orange color fruits in outdoor conditions for robotic applications", in *2016 Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, 2016, pp. 1–5.

[5] L. Bicker, *Coronavirus in south korea: How trace, test and treat may be saving lives*, Mar. 2020. [Online]. Available: https://www.bbc.com/news/world-asia-51836898.

[6] M. Fisher and C. Sang-hun, *How south korea flattened the curve*, Mar. 2020. [Online]. Available: https://www.nytimes.com/2020/03/23/world/asia/coronavirus-south-korea-flatten-curve.html.

[7] R. Shoukrallah, "Road safety in five leading countries", *Journal of the Australasian College of Road Safety*, vol. 19, no. 1, pp. 9–12, 2008.

[8] M. Rumman, A. N. Tasneem, S. Farzana, M. I. Pavel, and M. A. Alam, "Early detection of parkinson's disease using image processing and artificial neural network", in *2018 Joint 7th International Conference on Informatics, Electronics Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision Pattern Recognition (icIVPR)*, 2018, pp. 256–261.

[9] *How do night vision goggles work?* [Online]. Available: https://wonderopolis.org/wonder/how-do-night-vision-goggles-work.

[10] D. D. Pukale, S. G. Bhirud, and V. D. Katkar, "Content-based image retrieval using deep convolution neural network", in *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, 2017, pp. 1–5.

[11] D. Barik and M. Mondal, "Object identification for computer vision using image segmentation", in *2010 2nd International Conference on Education Technology and Computer*, vol. 2, 2010, pp. V2-170-V2–172.

[12] A. B. Abche, F. Yaacoub, A. Maalouf, and E. Karam, "Image registration based on neural network and fourier transform", in *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, 2006, pp. 4803–4806.

[13] F. Tanriverdi, D. Schuldt, and J. Thiem, "Hyperspectral imaging: Color reconstruction based on medical data", in *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, 2018, pp. 194–199.

[14] C. Woodford, *How does night vision work?*, Jun. 2019. [Online]. Available: https://www.explainthatstuff.com/hownightvisionworks.html.

[15] M. Lindenbaum, M. Fischer, and A. Bruckstein, "On gabor's contribution to image enhancement", *Pattern Recognition*, vol. 27, no. 1, pp. 1–8, 1994.

[16] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images", in *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, IEEE, 1998, pp. 839–846.

[17] L. T. Bang, W. Li, M. Piao, M. A. Alam, and N. Kim, "Noise reduction in digital hologram using wavelet transforms and smooth filter for three-dimensional display", *IEEE Photonics Journal*, vol. 5, no. 3, pp. 6 800 414–6 800 414, 2013.

[18] Y. Wang and H. Zhou, "Total variation wavelet-based medical image denoising", *International Journal of Biomedical Imaging*, vol. 2006, 2006.

[19] D. Gerónimo, J. Serrat, A. M. López, and R. Baldrich, "Traffic sign recognition for computer vision project-based learning", *IEEE Transactions on Education*, vol. 56, no. 3, pp. 364–371, 2013.

[20] Z. Cai and L. Shao, "Rgb-d data fusion in complex space", in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 1965–1969.

[21] S. Kim and N. I. Cho, "A lossless color image compression method based on a new reversible color transform", in *2012 Visual Communications and Image Processing*, 2012, pp. 1–4.

[22] M. Farenzena and A. Fusiello, "Recovering intrinsic images using an illumination invariant image", vol. 3, Jan. 2007, pp. 485–488. DOI: 10.1109/ICIP.2007.4379352.

[23] I. D. Tithi, U. S. K. Shuchi, N. A. Tasneem, M. I. Mobin, and M. A. Alam, "Brain fmri image classification and statistical representation of visual objects", in *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 2019, pp. 1–6.

[24] I. Benkhaled, I. Marc, and D. Lafon, "Colour contrast detection in chromaticity diagram: A new computerized colour vision test", in *2017 IEEE Western New York Image and Signal Processing Workshop (WNYISPW)*, 2017, pp. 1–4.

[25] B. Zhang, "Computer vision vs. human vision", in *9th IEEE International Conference on Cognitive Informatics (ICCI'10)*, 2010, pp. 3–3.

[26] M. A. Alam, M. Piao, L. Gang, Y. Piao, and N. Kim, "Three dimensional projection-type integral imaging display system using directional projection and elemental image resizing method", in *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*, 2013, pp. 1–4.

[27] J. Yoo, Y. Ha, and S. K. Subhashdas, "Color constancy method based on local chromaticity distribution and illuminant influence for hue angle", in *2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)*, 2017, pp. 1–5.

[28] W. Khan, E. Phaisangittisagul, L. Ali, D. Gansawat, and I. Kumazawa, "Combining features for rgb-d object recognition", in *2017 International Electrical Engineering Congress (iEECON)*, 2017, pp. 1–5.

[29] S. Bianco, C. Cusano, and R. Schettini, "Single and multiple illuminant estimation using convolutional neural networks", *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4347–4362, 2017.

[30] P. K. Nigam and M. Bhattacharya, "Colour vision deficiency correction in image processing", in *2013 IEEE International Conference on Bioinformatics and Biomedicine*, 2013, pp. 79–79.

[31] X. Ye, L. Wang, H. Xing, and L. Huang, "Denoising hybrid noises in image with stacked autoencoder", in *2015 IEEE International Conference on Information and Automation*, IEEE, 2015, pp. 2720–2724.

[32] E. S. Dunstone, "Image processing using an image approximation neural network", in *Proceedings of 1st International Conference on Image Processing*, vol. 3, 1994, 912–916 vol.3.

[33] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of psnr in image/video quality assessment", *Electronics letters*, vol. 44, no. 13, pp. 800–801, 2008.

[34] S. Park, S. Yu, M. Kim, K. Park, and J. Paik, "Dual autoencoder network for retinex-based low-light image enhancement", *IEEE Access*, vol. 6, pp. 22 084–22 093, 2018.

[35] E. H. Land and J. J. McCann, "Lightness and retinex theory", *Josa*, vol. 61, no. 1, pp. 1–11, 1971.

[36] Y. Hashisho, M. Albadawi, T. Krause, and U. F. von Lukas, "Underwater color restoration using u-net denoising autoencoder", in *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)*, IEEE, 2019, pp. 117–122.

[37] 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.

[38] O. DeGuchy, F. Santiago, M. Banuelos, and R. F. Marcia, "Deep neural networks for low-resolution photon-limited imaging", in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 3247–3251.

[39] D. Sugimura, T. Mikami, H. Yamashita, and T. Hamamoto, "Enhancing color images of extremely low light scenes based on rgb/nir images acquisition with different exposure times", *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3586–3597, 2015.

[40] K. He, J. Sun, and X. Tang, "Guided image filtering", in *European conference on computer vision*, Springer, 2010, pp. 1–14.

[41] Q. Yan, X. Shen, L. Xu, S. Zhuo, X. Zhang, L. Shen, and J. Jia, "Cross-field joint image restoration via scale map", in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1537–1544.

[42] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines", in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[43] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning", in *International Workshop on Artificial Neural Networks*, Springer, 1995, pp. 195–201.

[44] L. Gondara, "Medical image denoising using convolutional denoising autoencoders", in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, IEEE, 2016, pp. 241–246.