LOW-LIGHT IMAGE ENHANCEMENT WITH RESNET ARCHITECTURE AND SELF-CALIBRATED ILLUMINATION NETWORK

ZAYAR TUN

M.C.Tech. DECEMBER 2022

LOW-LIGHT IMAGE ENHANCEMENT WITH RESNET ARCHITECTURE AND SELF-CALIBRATED ILLUMINATION NETWORK

BY

ZAYAR TUN

B.C.Tech.

A dissertation submitted in partial fulfillment of the requirements for the degree of

Master of Computer Technology

(M.C.Tech.)

University of Computer Studies, Yangon

DECEMBER 2022

ACKNOWLEDGEMENTS

I would like to give this opportunity for me to express my sincere thanks to people who have assisted and helped me with various aspects of writing this thesis.

First and foremost, I would like to express my sincere thanks to **Prof. Dr.Mie Mie Khin,** Rector, University of Computer Studies, Yangon, for overall her kind permission to submit my thesis.

My heartfelt thanks and respect go to **Prof. Dr. Thandar Win**, Principal and Pro-rector of Computer University (Myeik), for her invaluable and administrative support.

I owe my special sincere thanks and regards go to **Dr. Htar Htar Lwin**, Pro-rector, Head of Faculty of Computer Systems and Technologies, University of Computer Studies, Yangon, for her giving me a lot of suggestions, kind of management, encouragement and motivation on my thesis work to reach the success of goal.

I would like to express my deeply thanks to **Dr. Amy Tun, Professor**, Course coordinator of Master's (CT), University of Computer Studies, Yangon, for her suggestion and encouragement throughout the completion of my thesis.

I would like to express my deepest gratitude and regards go to my supervisor, **Dr. Khant Kyawt Kyawt Theint**, Lecturer, Faculty of Computer Systems and Technologies, University of Computer Studies, Yangon, for her guidance, advice, support, supervision and encouragement during the period of study towards completion of my thesis.

I also wish to express my respectful gratitude to Daw Hnin Yee Aung, Lecturer, Department of English, University of Computer Studies, Yangon, for her editing this thesis from the language point of view.

Moreover, I wish to extend my thanks to all my teachers who taught me throughout the master's degree course and my friends for their advice and cooperation.

Finally, I especially deepest thank to my father, my mother and my family for their unconditional love, support, help and encouragement during my thesis.

ABSTRACT

Generally, low-light image enhancement techniques are mostly not just made to achieve both visual quality and computational efficiency but also commonly invalid in unknown complex scenarios. The system is focused on the image high quality displaying of low-light images using enhancement techniques. This system is used Self-Calibrated Illumination (SCI) module Network combination with Convolutional Neural Network (CNN) based on ResNet architecture Network to enhance the low-light image. In this system, Low-Light (LOL) dataset is applied. The system will be used LOL testing dataset for performance evaluation of the model. This system is implemented with the software program as Python language code and Anaconda application for running**.** Moreover, this system uses the three types of low-light image dataset for testing as LOL images, captured images by Camera and Black&White dataset.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATION

CHAPTER 1

INTRODUCTION

When one takes images in low-light states, the images commonly support from low visibility. The visual aesthetic of images, this low quality might greatly degenerate the performance of the many personal computer vision and multimedia system algorithms that are fundamentally designed for high-quality inputs. Low-light setting is connected integral a part of system everyday activities. As night-time, the quantity of available lightweight decreases, inflicting the environment to be darker and darker and later on touching the talents to perform even material tasks because of lack of visibility. In addition, people get low-light image in the process of image acquisition therefore the complexness of the natural environment and time. And then, low-light images usually get from two problems. First, they have low visibility that is small pixel values. Second, noise convert significant and reduce the image content, because of low signal to noise ratio. Therefore, low-light image techniques would be used.

The low-light image techniques described as followed:

- Low-Light Image Enhancement (LLIE)
- Using Dehazing Algorithm, Low-Light Image Enhancement (LLIE)
- Using Imreducehaze Optional Parameters, Improving Results Further
- Improving Low-Light Image of Another Example
- Using Different Color Space, Reducing Color Distortion
- Using Denoising, Improving Results
- Illumination Map Estimation
- Controlling

There are different methods proposed by various researchers until now which build Retinex for image contrast enhancement. Low-light image enhancement usually formal images captured in low-light quality like night time, where the usual goal is to illumination and improve the contrast of the image for more visual quality and display details that were unseen in night. In addition, methods of low-light image enhancement are Median filtering, Noise removal using Wiener, Unsharp mask filtering, and so on. The goal of low-light image enhancement is to increase the light and clarity of image. Therefore, people provide better useful. And then, building a deep convolutional neural network model is applied ResNet architecture and Self-Calibrated Illumination (SCI)

Networks for fast, adjustable, and powerful better illumination images in real-world low-light scenarios. The Retinex theory [14], the system defines a product of illumination and reflectance from the object. Retinex targets on active dimension and color stability of an image. There are different methods proposed by many researchers until now which use Retinex for image contrast enhancement. The low-light image has weakness of low contrast, not even illumination and low image visibility as a result of the lacking of light source, which is greatly bad for the next processing of images like feature extraction and image segmentation. There have been a set of improvements in the field of Deep Learning and Computer Vision [14]. Usually with the introduction of more deep convolutional neural networks, these models helped execute state-of-the-art results on problems like image recognition and image classification.

Therefore, for many years, the deep learning architectures became better layers to solve better complex tasks which also helped in increasing the performance of classification and recognition tasks and also making them robust. And the system is using build a deep convolutional neural network model to increase low-light images. In addition, the system learns combination of a new Self-Calibrated Illumination (SCI) learning framework for quick, adjustable, and powerful brightening images in real-world low-light actions. This system also gets a cascaded illumination learning process with weight sharing to take this task. Considering the computational burden of the cascaded pattern. This system constructs the self-calibrated module which realizes the convergence between results of each stage, producing the gains that use the single basic block for inference.

1.1 Objectives of the Thesis

The system study has aimed to achieve the following main objectives:

- To enhance the contrast and illumination of low-light images automatically
- To obtain better interpretation and visualization effects
- To apply self-calibrated learning framework in low-light images enhancement process to be quick, adjustable, and powerful brightening
- To provide the low-light images more clear viewing during

1.2 Motivation of the System

Being low-light image enhancement techniques are often similarly hard to control both visual quality and computational efficiency. When one takes images in low-light states, the images normally stand from low visibility. This poor quality could great reduce the performance of the many computers vision and multimedia system algorithms that are especially designed for high-quality input of images. These images will have poor active ranges with high noise levels that have an effect on the performance of computer vision algorithms. To build computer vision algorithms powerful in low-light states, low-light image enhancement to increase the visibility of an image is the require period. The purpose of image enhancement is to increase the interpretability or perception of information in images for human observers, or to support more input for other automated image processing techniques. The image enhancement system consists about interesting concepts and knowledge in this area of image enhancement. Therefore, this point is motivation to do the low-light image enhancement system for thesis.

1.3 Organization of the Thesis

The organization of the thesis consists five chapters. Chapter 1 describes the introduction of low-light image enhancement, objectives, motivation of the system, and organization of the thesis. Chapter 2 describes background theory of the system, digital image processing, deep learning and the basic concept of neural network. Chapter 3 describes convolutional neural network, CNN based on ResNet model of image enhancement, the calibration, and self-calibrated convolution, and learning of self-calibrated illumination network. Chapter 4 describes the proposed system flow, low-light image of dataset, the implementation software of the system, and the experiment results. Finally, Chapter 5 describes the conclusion, advantages of the system, limitations of the system and further extensions of the system.

CHAPTER 2

BACKGROUND THEORY

In this chapter, the background theory of introduction of deep learning, the basic concepts of neural network, image enhancement, low-light image enhancement, and self-calibration are described.

2.1 Deep learning

Deep Learning is a tree of machine learning. Opposite traditional machine learning algorithms, various of which have a finite capacity to train no problem how much data they acquire. The deep learning system increase their performance with permission to better data and the machine version of better skills. Deep learning manages various Artificial Intelligence (AI) applications and services. This deep learning increase automation, performing logical and physical tasks without person interruption. Deep learning defined as a machine learning technique. When machines have learned acceptable skills pass deep learning, they perhaps place to work for particular tasks like driving a car, detecting weeds, inspecting machinery, detecting diseases, etc. In moreover, Deep learning technique lies daily products and services. Deep learning is a branch of machine learning, which is usually a neural network with three or more layers. These neural networks aim to act the action of a person brain, permission it to learn from many data. When a neural network with one only layer can stable make nearly predictions, combination hidden layers can support correct and better for accuracy.

Deep learning trains computers to do what arrives basically to people. Deep learning is obtaining endless application just now and for great reason. It is obtaining results that were impossible before. In deep learning, a computer model trains to perform classification tasks exactly from images, text, or sound. Deep learning has arrived just as an effective tool for analyzing large data. Deep learning uses multiplex algorithms and artificial neural networks to train machines or computers in order that they can learn from experience, classify and recognize data or images just like a human brain does. Within Deep learning, a Convolutional Neural Network (CNN) [2][17] defined a type of artificial neural network, which is used for image or object recognition and classification.

2.1.1 Deep Learning Works

Deep learning neural network, aims to the person brain pass an addition of data inputs, weights and bias. These components work with to accurately accept, classify, and display describing the object. Next, other process called backpropagation uses algorithms such as low gradient, calculate errors and controls the weights and train the model. Deep learning algorithms are complex and different types of neural networks to different types of problems or datasets. Some of the real-world deep learning applications are law enforcement, Healthcare, Financial services and Customer service. Figure 2.1 is showing the structure of Artificial Intelligence, Machine Learning and Deep Learning.

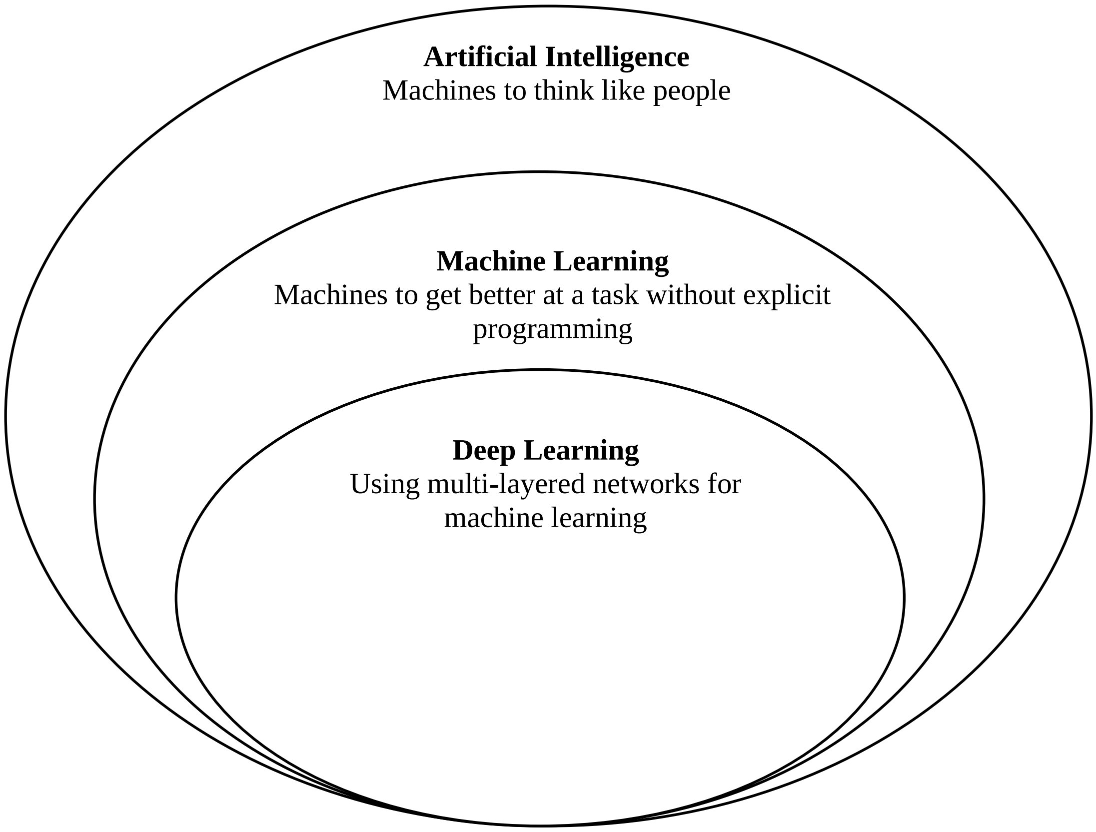


Figure 2.1 The Structure of AI, Machine Learning and Deep Learning

Today, the future of deep learning are different neural network architectures enhanced for types of input and tasks. Deep learning architectures are Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). CNN is better at images classification and RNN is process serial data. These two network models execute what is known as controlled learning. Recent artificial neural networks were based on 1950s deciding of how person brains process information. In addition, CNNs used in computer vision and image classification applications such as images detect features and patterns. RNNs used in natural language and speech recognition applications such as it influences time series data.

2.2 The Basic Concepts of Neural Network

One of the basic understandings one needs to have about Artificial Intelligence (AI) is that AI systems are built to perform tasks that normally demands human intelligence. The idea here is to use the model of a human brain to build an artificial intelligence. A neural network is a model designed to imitate the learning process of the human brain. Deep learning is a method that gives the neural network the opportunity to figure out, the important features in a given data. A neural network uses the following step to learn how to make predictions. The steps are as follow:

1. Receives data as inputs.
2. Generates an output as a prediction.
3. Compares the prediction with the desired result.
4. Use the outcome of the comparison to modify or adjust its internal parameters to make a better prediction next time

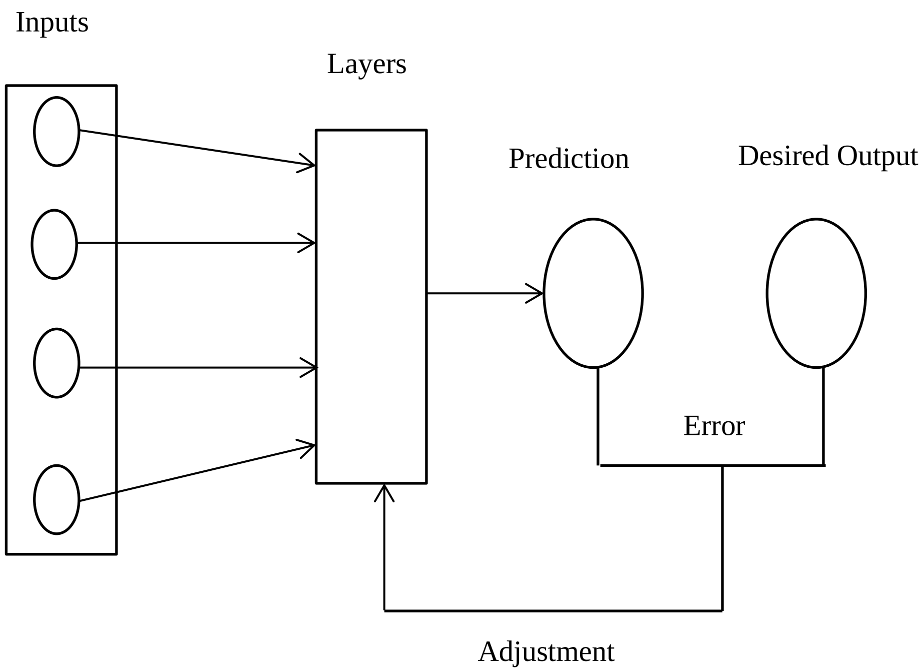


Figure 2.2 A neural network uses step by step to learn how to make predictions

Figure 2.2 is a neural network uses step by step to learn how to make predictions. The neural network is like a layer’s box that consists of serval layers. Each layer requires input data to be in vector forms. Input data for the neural networks can come in any form or kind (images, texts, etc.), this is one very good advantage of using neural networks. The python NumPy library enables us to store vectors as arrays. Each layer does some sort of filtering or feature engineering process on the input data.

2.2.1 Neural Network

A neural network is defined as a software solution that leverages machine learning (ML) algorithms to mimic the operations of a human brain. Neural networks process data more efficiently and feature improved pattern recognition and problem-solving capabilities when compared to traditional computers. Neural networks are also known as artificial neural networks (ANNs) or simulated neural networks (SNNs). Neural networks are a subtype of machine learning and an essential element of deep learning algorithms. Just like its functionality, the architecture of a neural network is also based on the human brain. It is highly interlinked structure allows it to imitate the signaling processes of biological neurons.

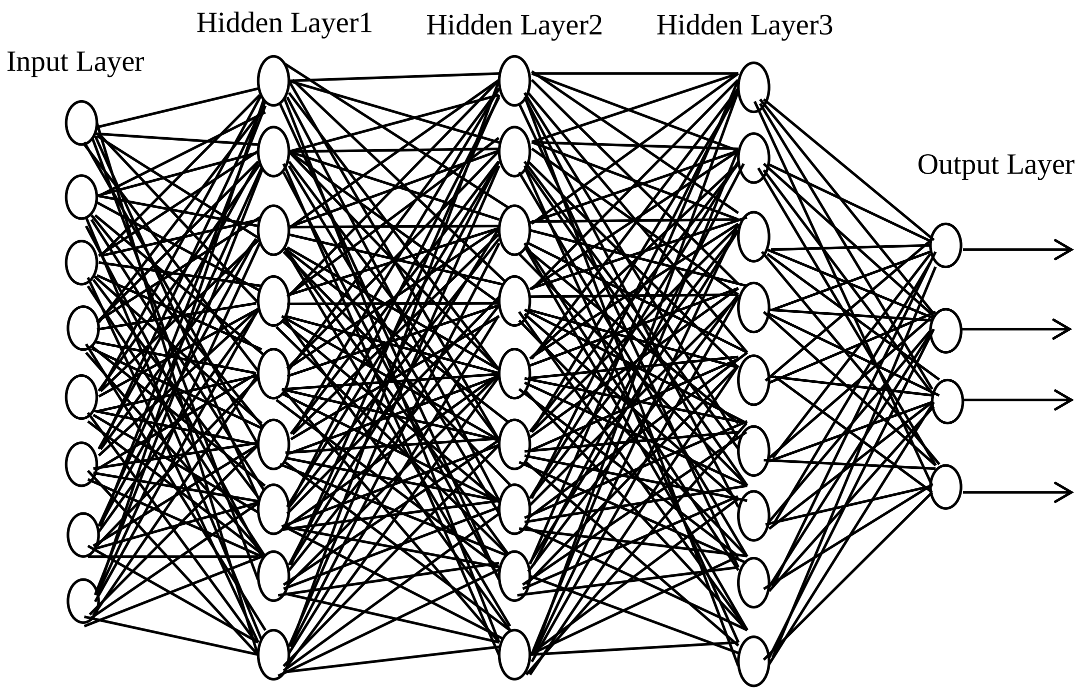


Figure 2.3 The Architecture of a Neural Network

Figure 2.3 is the architecture of a neural network comprises nodes layers. If the threshold value of the node is not crossed, data is not transferred to the next network layer. Unlike traditional computers, which process data sequentially, neural networks can learn and multitask. It can be said that neural computers program themselves to drive solution to previously unseen problems. Additionally, traditional computers operate using logic functions based on a specific set of calculations and rules. Conversely, neural computers can process logic functions and raw inputs such as images, videos, and voice. Neural networks are capable of classifying and clustering data at high speeds. This means, among other things, that they can complete the recognition of speech and images within minutes instead of the hours that it would take when carried out by human experts. The most commonly used neural network today Google search algorithms.

2.2.2 Neural Network Works

The ability of a neural network to think has revolutionized computing as the system know it. These smart solutions are capable of interpreting data and accounting for context. For critical steps that neural networks take to operate effectively are as follow:

i. Associating or training enables neural networks to remember patterns. If the computer is shown an unfamiliar pattern, it will associate the pattern with the closest match present in its memory.
ii. Classification or organizing data or patterns into predefined classes.
iii. Clustering or the identification of a unique aspect of each data instance to classify it even without any other context present.
iv. Prediction or the production of expected results using a relevant input, even when all contexts are not provided upfront.

Neural networks require high throughput to carry out these functions accurately in near real-time. This is achieved by deploying numerous processors to operate parallel to each other, which are arranged in tiers. The neural networking process begins with the first tier receiving the raw input data. The system can compare this to the optic nerves of a human being receiving visual inputs. After that, each consecutive tier gets the results from the preceding one. This goes on until the final tier has processed the information and produced the output. Neural network performs by simulating a large number of inter connected processing units that are resemble conclusion about all versions of the neurons. A neural network is a process or method in artificial intelligence that trains or learns computers to process data in a way.

2.2.3 Types of Neural Network

Neural Networks are classified based on several factors, including their depth, the number of hidden layers, and the I/O capabilities of each node. The six key types of neural networks are as follow:

1. Convolutional Neural Networks
2. Deconvolutional Neural Networks
3. Recurrent Neural Networks
4. Feed-forward Neural Networks
5. Modular Neural Networks
6. Generative Adversarial Network

Convolutional neural networks are being a highly popular neural networking model, convolutional neural networks leverage a type of multilayer perceptron and include one or more convolutional layers. These layers can be either pooled or entirely connected. This neural networking model uses principles from linear algebra, especially matrix multiplication, to detect and process patterns within images. The convolutional layers in this model can create feature maps that capture a specific area within a visual input. The site is the broken down further and analyzed to generate valuable outputs. Convolutional neural networks are beneficial for AI-powered image recognition applications. This type of neural network is commonly used in advanced use causes such as facial recognition, natural language processing (NLP), optical character recognition (OCR), and image classification. It is also deployed for paraphrase identification and signal processing.

Deconvolutional neural networks work on the same principles as convolutional networks, except in reverse. This specific application of AI aims to detect lost signals or features that many have previously been discarded as unimportant as the convolutional neural network was executing its assigned task. Deconvolution neural networks are helpful for various applications, including image analysis and synthesis. This complex neural network model works by saving the output generated by its processor nodes and feeding them back into the algorithm. This process enables recurrent neural networks to enhance their prediction capabilities. In this neural network model, each node behaves like a memory cell. These cells work to ensure intelligent computation and implementation by processing the data they receive. However, what sets this model apart is its ability to recollect and reuse all processed data. A strong feedback loop is one of the critical features of a recurrent neural network. These neural network solutions can self-learn from their mistakes. If an incorrect prediction is made, the system learns from feedback and strives to make the correct prediction while passing the data through the algorithm the second time. Recurrent neural networks are commonly used in text-to-speech applications. This simple neural network variant passes data in a single direction through various processing nodes until the data reaches the output node. Feed-forward neural networks are designed to process large volumes of noise data and create clean outputs. This type of neural network is also known as the multi-layer perceptron (MLP) model.

Feed-forward neural network architecture includes the input layer, one or more hidden layers, and the output layer. Despite their alternate name, these models leverage sigmoid neurons rather than perceptron, thus allowing them to address nonlinear, real-word problems. Feed-forward neural networks are the foundation for facial recognition, natural language processing, computer vision, and other neural network models. Modular neural networks feature a series of independent neural networks whose operations are overseen by an intermediary. Each independent network is a module that uses distinct inputs to complete a particular part of the larger network's overall objective. The modules do not communicate with one another or interfere with each other's processes while computation occurs. This makes performing extensive and complex computational processes more efficient and quicker. Generative adversarial networks are a generative modeling solution that leverages convolutional neural networks and other deep learning offerings to automate the discovery of patterns in data. Generative modeling uses unsupervised learning to generate plausible conclusions from an original dataset. Generative adversarial networks train generative models by creating a supervised learning problem containing a generator model and a discriminator model. The former is prepared to develop new conclusions from the input.

2.3 Image Enhancement

Image enhancement is the process of automatically operating a kept image using operating system of enhancement image. The machines used for image enhancement with various operation system software like image editors, filters, and other machines for different properties of all image. Image enhancement is the resolution of increasing the quality and original information content of data before image processing. This image processing consists of contrast enhancement, density slicing, spatial filtering and so on. Contrast enhancement is also known as stretching, executed by linear transformation paying the original gray level. And then, density slicing changes the stable gray range to a sequence of density intervals. A separate color or different features of symbol is marked these density intervals. Next, spatial filtering is the basically appearing linear features. Examples and some methods of image enhancement are described in Figure 2.4,2.5 and 2.6.

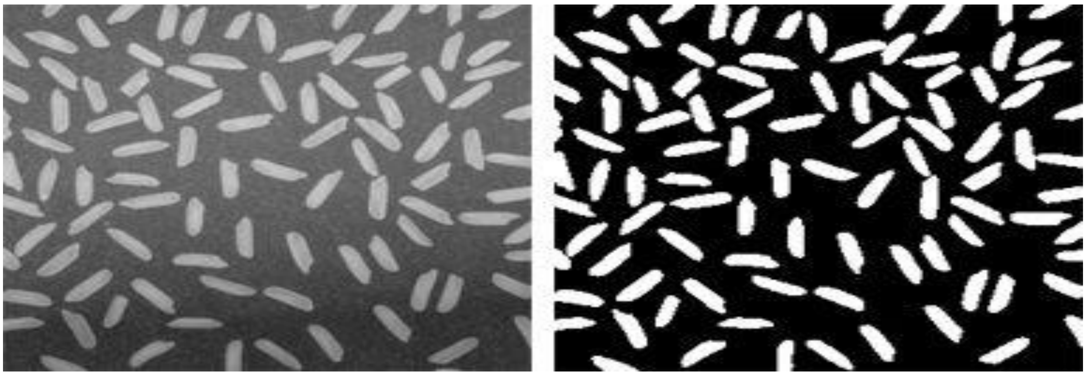


Figure 2.4 Image Enhancement and correcting nonuniform Illumination with morphological operators

Figure 2.5 Image Enhancement and enhancing grayscale images with histogram equalization

Figure 2.6 Image Enhancement and deblurring images using Wiener filter

Image enhancement defines the process of featuring specific information of an image, together with decreasing or reducing some needless information in keeping with specific needs. Examples such as disclosing burred details, eliminating noise, and adjusting highlight features levels of an image.

2.3.1 Image Enhancement Techniques

Image enhancement techniques are described into two categories:

- Spatial Domain: Enhancement of the image space, spatial domain methods execute processes on pixels directly.
- Frequency Domain: Enhancement gained from the Fourier transform; pixels are processed in indirectly the frequency domain.

Some of the applications of image enhancement are brighten image, contrast adjustment, deblur images, smooth and sharpen. Grayscale image of histogram equalization defines the transformation where output image has nearly the equal number of pixels at each gray level. Noises reducing are defined images at the bit of take from cameras. Noises reducing techniques decrease the resolution of noises with using linear or non-linear filters of smoothing image. Image smoothing is digital image processing technique that decreases and contains image noises. Smoothing filters consists of gaussian smoothing, average smoothing, and adaptive smoothing. Image sharpening filters feature edges by removing or reducing blur.

2.4 Low-Light Image Enhancement

Low-Light Image Enhancement (LLIE) purposes at increasing the viewpoint or realization of an image take in an environment with low-light conditions. When one take image during low-light states, the images suffer from poor visibility. In addition, this poor-quality image may completely degenerate the performance of various computer vision and multimedia algorithms. Therefore, the system proposes effective of low-light image enhancement (LIME) method [4]. Moreover, the illumination of each pixel is evaluated by detecting the maximum value in R, G and B channels. This system enhances the illumination map by moving structure prior on it. Low light image enhancement is defined to be the complex of operation tasks in an image processing. When the images are captured low light, the quality of images are poor illumination conditions. Figure 2.7 is Before and After enhancement of low-light image using python and deep learning.

Before Enhancement **After Enhancement**

Figure 2.7 Example of Low-Light Image Enhancement using python and deep learning

2.4.1 Low-Light Image Enhancement Methods

Low-Light Image Enhancement usually classified into the ratio of cognitive modelling, contrast improvement, and illumination correction methods. Cognitive modelling-based methods accurate low-light and removed color signals by dividing acquired image into illumination and coefficient of reflection components using Retinex theory. Histogram equalization used to increase the contrast ratio, and gamma correction to increase the brightness information of image. Although, these methods have weakness in performance. Because they use operating methods besides considering the illumination component of an image.

Deep learning techniques applied the low-light image enhancement problem. Retinex theory based, adjustment pipeline, and encouraging performance. Although, Retinex based deep learning are insufficient, avoid various useful methods.

2.5 Self-calibration

Self-calibration differs in the amount of period it takes on various devices. A camera could be calibrated using point matches between images, and termed the methods self-calibration. Some methods self-calibrate directly in one step, while other uses a stratified approach and calibrate via a projective. The objective of self-calibration is the same for daily device. Self-calibration can build without external connections.

This resolution consists of routing known internal reference voltage to all channels of the board. The voltage of reference is read at various gain settings and compared to the expected value. This voltage is temperature secured and is indicated like way to compensate for temperature changes. Self-calibration also known as a process performed by a person for the objective of making an inspection, measuring, test equipment. The process may be needed at intervals like power on sequence.

In addition, a point x in one image achieves a line in the other on which its corresponding point new x must be lie. With two views, the two camera coordinate systems are connected by a rotation R and a translation T. The system of self-calibration proposes to control or management of convolutional feature transformation in two various scale spaces. An original scale space in which feature maps contribution or distribute the same of resolution with the input and a less latent space after down-sampling.

2.5.1 Self-Calibrated Convolutions

The image feature of transformation process is executed in achieve or complete equivalent extension. The system of the outputs from one and all extension are connected like the final output. Like group of convolutions, the proposed self-calibrated convolutions break the learnable convolutional filters to various of portions, until now individually, each portion of filters is individually prepared however ability for a particular performance or process. Self-calibrated convolution constructs spatial and outside channel approximately each spatial location passes a self-calibration operation. Therefore, it is support CNN achieve better discriminative representations

CHAPTER 3

METHODOLOGY

In this chapter, concepts and details of convolutional neural networks (CNNs), residual networks (ResNets) of deep learning, self-calibrated module and learning of self-calibrated illumination (SCI) network will be presented in detail.

3.1 Concepts of Convolutional Neural Networks

The concept of convolutional neural networks in deep learning has appeared much above the past less ten years due to effectively addressing great datasets. And making the methodology computer systems accomplished sufficient to solve computational problems. The system of hidden layers initiates a new epoch, with an ancient technique existing non efficient, exceptionally when it arrives to difficult such as Recognition of pattern, Detection of the object, Segmentation of an image, and other image processing based difficult of problems. Convolutional neural network is one of the highest situated deep learning neural networks. The first convolutional neural network as history and background were developed around the 1980s. It was construct to digitals of handwritten cognization. The neural network architecture was direct, with five layers of 5x5 convolutional layers and 2x2 max pooling. It was defined name as LeNet. However, the use of convolutional neural network limited due to different problem and reasons, like the require for many training data and operational resources.

In the deep learning, convolutional neural network is the class of deep neural networks, which was being highest deployed in the image recognition. Convolutional Neural uses a special method which is being known as the convolution. The mathematical of convolution is a computational operation being adjusted on two functions. In addition, convolutional neural network (CNN) [2][17] is a type of multidimensional neural network. CNN is also a deep learning architecture. The CNN is powerful for various fields of computer vision and regular language processing. The main target of all of the basic CNN components give a general view of CNN. CNN is also a type of deep learning model for processing. It has a network pattern like images, which is designed to automatically learn spatial hierarches of features from low level pattern to high level pattern.

3.1.1 Convolution Neural Network (CNN/ ConvNet)

Artificial Intelligence has been viewing improvement between the capabilities of people and machine. A convolutional neural network is deep learning algorithm that take in image, learnable weights and biases to different objects in the image. The preprocessing required in convolutional neural network of comparing to other classification algorithms. A convolutional neural network (CNN)[2][17] is neural network that has various convolutional layers. CNNs are used usually for image processing, image segmentation, image classification and other auto correlated data. The most use of CNN is image classification. CNNs have been used for recognizing in natural language processing and recognition of speech, however frequently for natural language processing of recurrent neural networks (RNNs) are used.

CNN can be executed like U-Net architecture, which are approximately two reflected CNNs result in a CNN whose architecture can be showed in U shape. U-Nets are used the output requires to be of same size to the input like segmentation and image improvement. Convolutional neural networks are collected of various layers of artificial neurons. Where artificial neurons, a limitation of their connected with the science of biology counter parts, are operational functions that operate the weighted resulting sum of multiple inputs and outputs and activation value. When the system input an image in CNN, each layer achieves various activation functions that are transferred on to the next layer. The first of layer generally removes the basic features like horizontal edges. This output is transferred on to the next layer which finds various complex features like corners edges. As the system drive deeper to the network it can recognize various complex features like faces, objects, etc.

Convolutional neural network or CNN is a group of machine learning. CNN is a type of network architecture for deep learning algorithms. It is one of the different kinds of artificial neural networks which are used for various applications and data formats. CNN is specially used for recognition of image and tasks that consists of processing of the pixel data. There are other kinds of neural networks in deep learning, however for recognizing and determining objects, CNNs are the network architecture of option. This makes them most acceptable for computer vision functions and for applications where recognition of object is fundamental like autonomous vehicles are also known as self-driving cars and automatic face recognition. Artificial neural networks or ANNs are defined a fundamental element of deep learning algorithms. One kind of an ANN is a recurrent neural network or RNN that uses time array data as input. It is acceptable for applications requiring natural language processing, image captioning, recognition of speech and translation of language.

CNN architecture is similar to the connectivity pattern of the people brain. Especially the brain contains billions of neurons, CNN has neurons fixed in specific way. CNN neurons are fixed such as the brain of the forward portion, the region responsible for the processing optical encouragement. This arrangement makes certain that the complete visual field is enhanced, therefore eliminating the little-by-little image processing problem of traditional neural networks, which must be maintained images in resolution of removing pieces. In comparison with the older network, CNN details better performance with inputs of image and also with audio inputs of signal. The CNN is also another kind of neural network that can crack key information in both period sets and data of image. Because, it is really important for relating tasks of image like recognition of image, classification of object and recognition of pattern. To recognize patterns inside an image, CNN importance fundamental truths from linear algebra like matrix multiplication. CNN can classify the data of signal and audio.

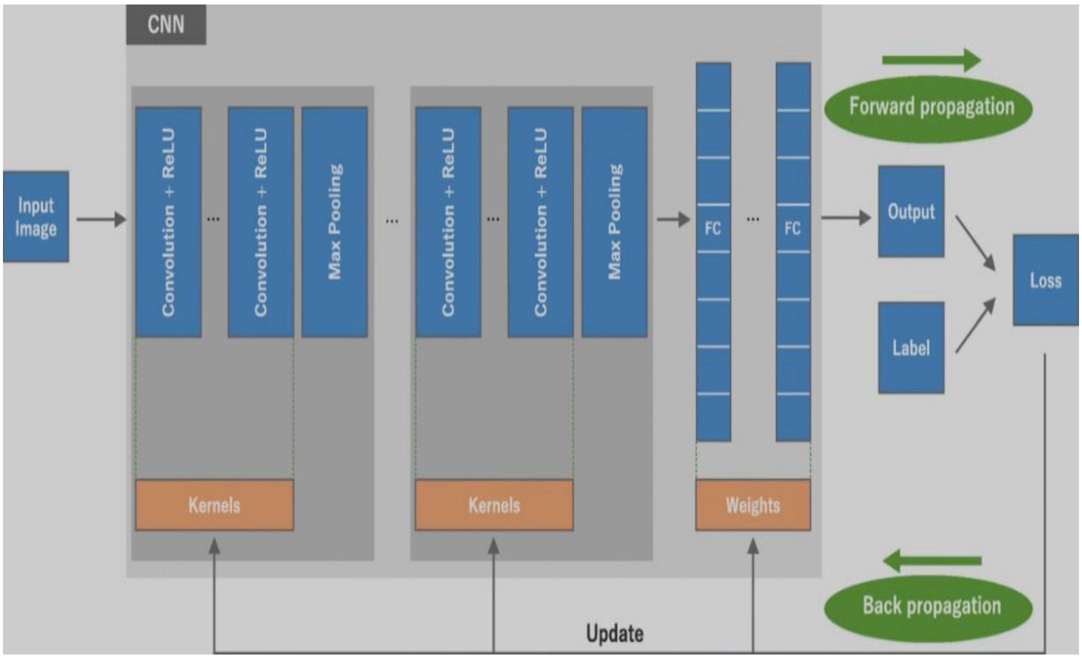


Figure 3.1 An overview of convolutional neural networks architecture and the training process

Figure 3.1 is an overview of convolutional neural networks (CNNs) architecture and the training process. A convolutional neural network is containing a layering of tree building blocks are such as first layer of convolution layers, next the pooling layers of example is max pooling, and the final layer of fully-connected (FC) layers. A model is performed below kernels and weights is operated with a loss function pass forward propagation on training dataset, and discoverable parameters that is kernels and weights are repaired with the loss value pass backward propagation using low gradient optimization algorithm

3.1.2 Basic Architecture of CNN

There are two main parts to CNN [2][17] architecture as presented as follow:

- A convolution tool different and identifies the different features of the image for evaluation in a process called Feature Extraction.
- The network of feature extraction consists of many pairs of convolutional or pooling layers.
- A fully connected layer makes use of the output from the convolution process and predicts the class of the image.
- This CNN model of feature extraction purposes to make smaller in size the number of features display in a dataset.

Figure 3.2 is two main parts to architecture of convolutional neural network (CNN).

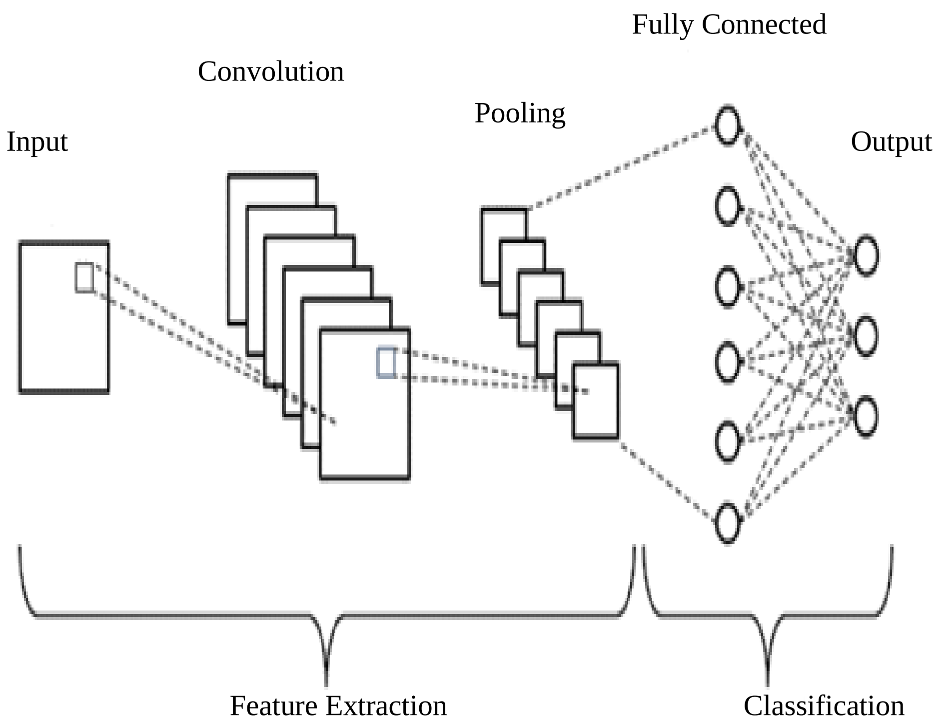


Figure 3.2 Two main parts to Architecture of CNN

3.1.3 Layer of CNNs (Deep Learning)

A deep learning convolutional neural network contains the three types of layers. There are (1) convolutional layer, (2) pooling layer, and (3) fully-connected layer. The convolutional layer is the first layer and fully-connected layer is last layer in the convolutional neural network. From the convolutional layer into the fully-connected layer, the convolution or complexity of the convolutional neural network increases or grows. It is this growing convolution that permits the convolutional neural to continually recognize larger portions and better convoluted features of an image before it finally recognizes the object in its totality.

A layer is a structure in the model architecture, which takes information from the previous layers and throughs it to the next layer. There are different layers in deep learning, specifically convolutional layer and maximum pooling layer in the convolutional neural network. Fully connected layer and ReLU layer in vanilla neural network. A convolutional neural network consists of an input layer, hidden layers and an output layer. In neural network, middle layers are called hidden and final convolution.

The hidden layers contain layers that perform convolutions. Commonly, this includes a layer that performs a dot product of the convolution kernel with the layer's input matrix. The activation function is commonly ReLU. As the convolutional kernel slides along the input matrix for the layer, the convolution operation generates a feature map, which in turn contributes to the input of the next layer. This is followed by other layers such as pooling layers, fully connected layers, and normalization layers.

3.2 Convolutional Layer

The convolutional layer is the building block of a CNN, and it is where the majority of computation occurs. It requires a few components, which are input data, a filter, and a feature map. The main of operations in the convolutional layer, which is the fundamental building block of a convolutional neural network. Second convolutional layer come after the initial convolutional layer. The system process of convolution requires a kernel or filter internal this layer working or moving pass the friendly fields of the image, establishing if feature is present in the image. More multiple iterations, the kernel flows more the complete image. After each iteration is operated between the input pixels and filter. The final output from the set of dots is defines like feature map or convolve feature. The image is converted to numerical value in this layer, which permits the convolutional neural network to describe the image and extract relevant patterns from it.

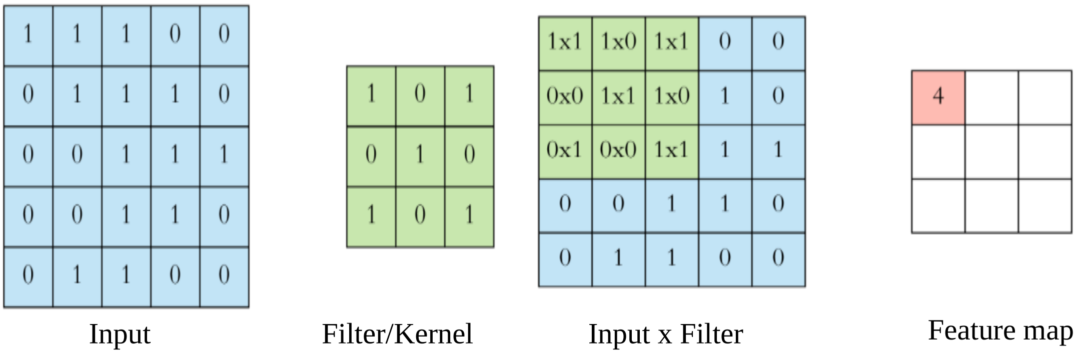


Figure 3.3 For example 5x5 input image, 3x3 convolutions using a convolution filter to produce a feature map

In Figure 3.3, first table of Input is the input to the convolution layer, for example 5x5 input image. Second table of Filter/Kernel is the convolution filter, also called the kernel, the system will use these terms interchangeably. This is called a 3x3 convolution due to the shape of the filter.

The system performs the convolution operation by sliding this filter over the input. At every location, the system does element-wise matrix multiplication and sum the result. This sum goes into the feature map. The green area where the convolution operation takes place is called the receptive field. Due to the size of the filter the receptive field is also 3x3.

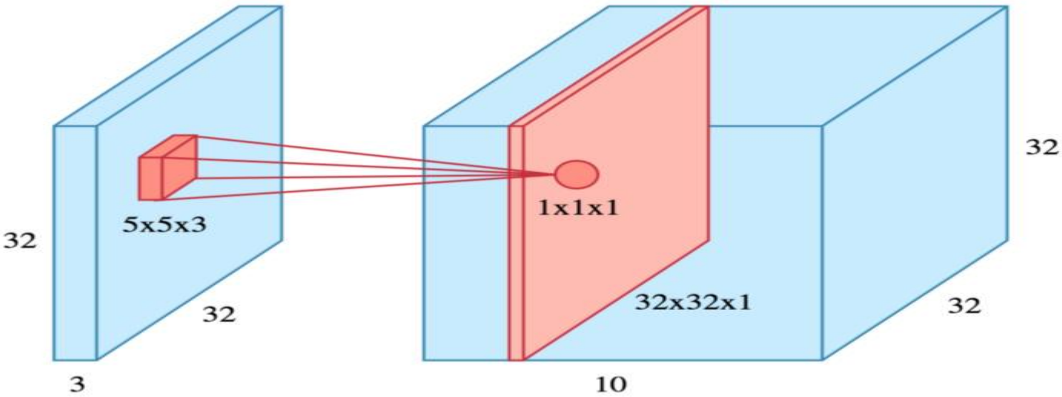


Figure 3.4 Example of 32x32x3 image and use a filter size 5x5x3

Figure 3.4 is a 32x32x3 image and use a filter of size 5x5x3. And then, Figure 3.5 is how two feature maps are stacked along the depth dimension.

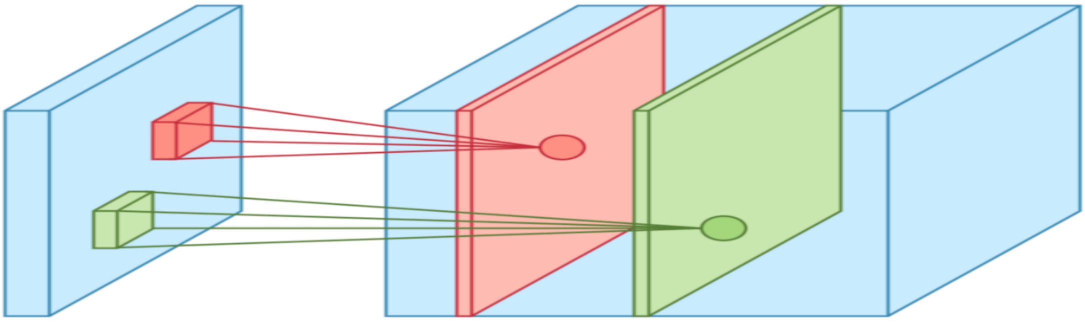


Figure 3.5 How two feature maps are stacked along the depth dimension

The convolutional layer is the core building block of a CNN, and it is where the majority of computation occurs. It requires a few components, which are input data, a filter, and a feature map are shown in Figure 3.6. Convolution Processing Equation:

$$g_{i,j}= \sum_{x=-m}^{m} \sum_{y=-m}^{m} f_{xy} x_{(i-x)(j-y)} \tag{3.1}$$

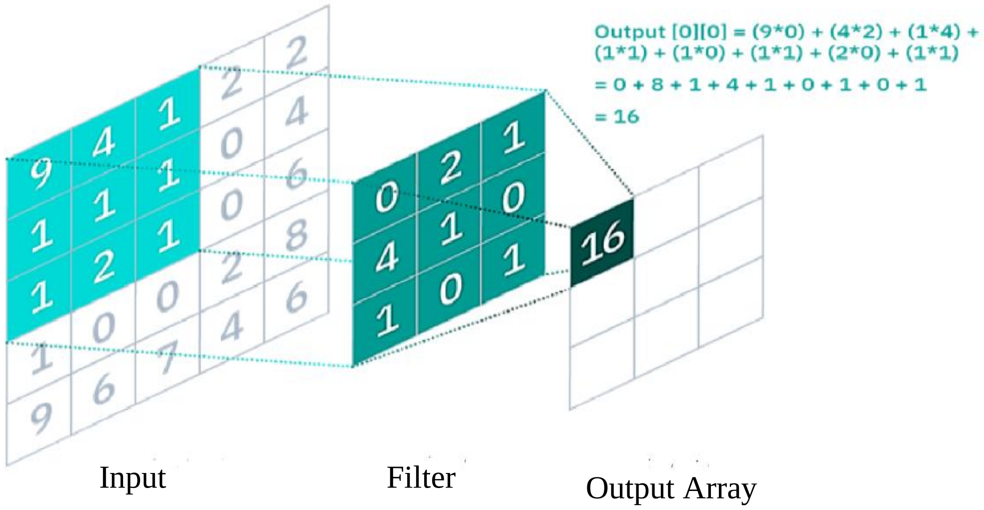


Figure 3.6 Sample majority of computation occurs. Few components are input data, a filter and a feature map

The convolution operation described enhancement the center of each kernel to element of the input tensor, and reduces the height and width of the output feature map compared to the input tensor. Padding, generally zero padding, is a technique to address this issue, where rows and columns of zeros are added on each side of the input tensor, so as to fit the center of a kernel on the outermost element and keep the same in-plane dimension through the convolution operation. Modern CNN [2][17] architectures usually employ zero padding to retain in-plane dimensions in order to apply more layers. Without zero padding, each successive feature map would get smaller after the convolution operation. The distance between two successive kernel positions is called a stride, which also defines the convolution operation. The common choice of a stride is 1, however, a stride larger than 1 is sometimes used in order to achieve down sampling of the feature maps. An alternative technique to perform down sampling is a pooling operation.

CNN model with regard to the convolution layer is to one and the same of the kernels that work best for a given task based on a given training dataset. Kernels are the only parameters automatically learned during the training process in the convolution layer; the size of the kernels, number of kernels, padding, and stride are hyperparameters that need to be set before the training process starts. Convolutional Neural Network of Layers, Parameters and Hyperparameters is shown in Table 3.1

Table 3.1 Convolutional Neural Network of Layers, Parameters and Hyperparameters

Layers of CNN	**Parameters of CNN**	**Hyperparameters of CNN**
Convolution Layer	Kernels	Size of Kernel, Number of kernels, Stride, Padding, Activation function
Pooling Layer	None	Filter size, Stride, Padding Pooling method
Fully-connected Layer	Weights	Activation Function and Number of Weights
Others		Model Architecture, Loss Function, Mini-Batch Size, Epochs, etc.

3.2.1 Non-linearity

Non-linear activation layers are contracted after all layers with weights. Therefore, this is also called learnable layers like fully-connected layers and convolutional layers in the convolutional neural network architecture. This non-linear conduct the activation layers and mapping of input to output will be non-linear. In addition, these layers give the convolutional neural network the potential to train extra complicated things. The activation function must have the ability to differentiate, which is especially significant feature, like it permits error back propagation to be used to train the network. Activation functions define non-linearity to the networks that is why the system call them non-linearities.

The system again passes the result of the convolution operation through ReLU activation function. The four activation functions are shown in Figure 3.7. For any kind of neural network to be powerful, it needs to contain non-linearity. Therefore, the values in the final feature maps are not actually the sums, but the ReLU function applied to them. Any type of convolution involves a ReLU operation, without achieving the network potential. The outputs of a linear operation such as convolution are then passed through a nonlinear activation function.

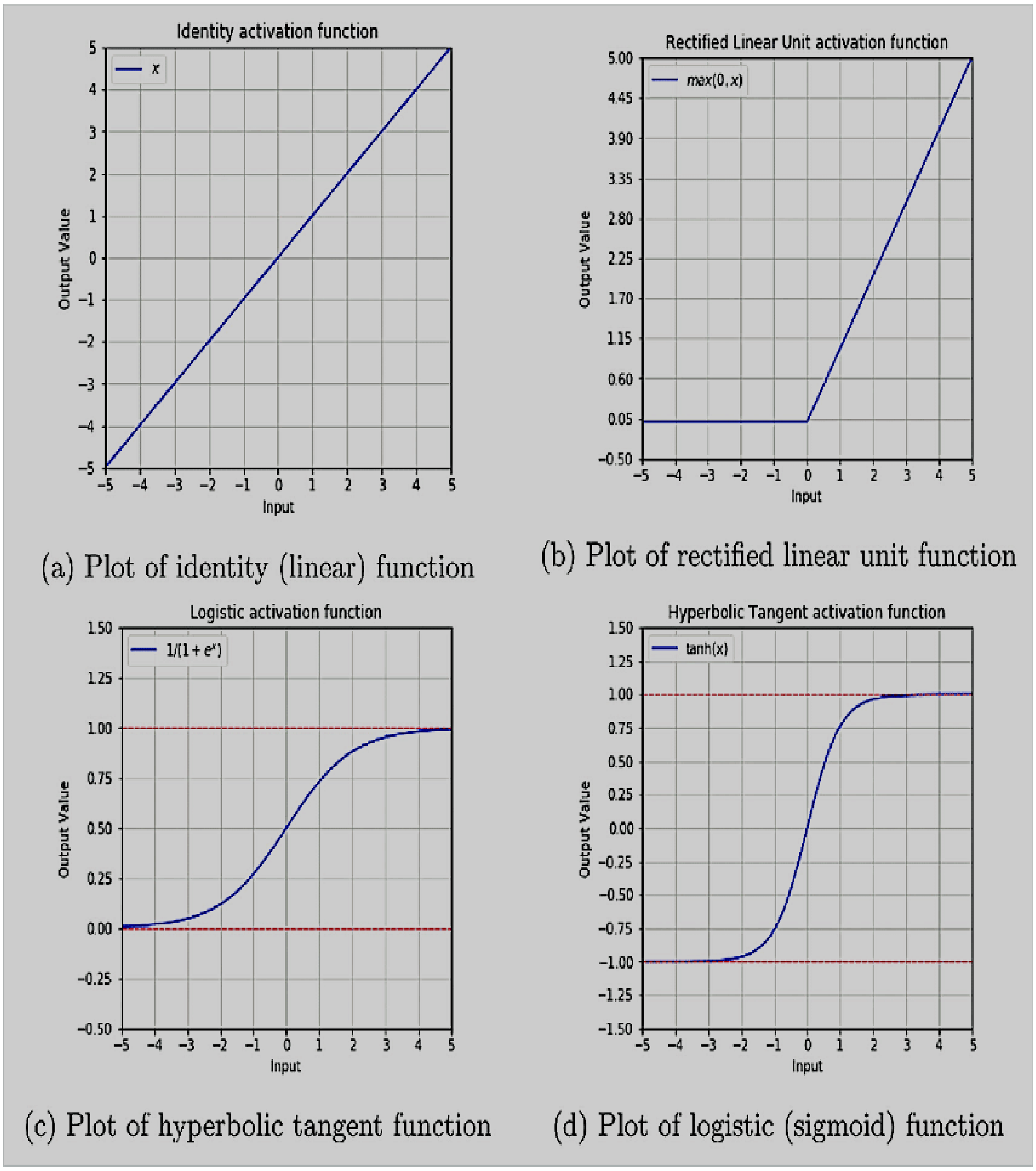


Figure 3.7 Activation Functions used: (a) Identity function, (b) ReLU function, (c) Tanh function and (d) Sigmoid function

However, smoothing of the nonlinear functions such as sigmoid or hyperbolic tangent (tanh) function, were used already or formerly because they are operational defining of a biological neuron action, the nonlinear activation function used is the rectified linear unit (ReLU), which simply calculates the function: $f(x) = \max(0, x)$. It learns 6 times exceeding of tanh. The system of output value will be zero when input value is lower than zero. If input is greater than or equal to zero, output is equal to the input. When the input value is positive, derivative is 1, hence there will be no squeezing effect which occurs in the case of backpropagation errors from the sigmoid function. Both the ANN and autoencoder the system saw before achieved this by passing the weighted sum of its inputs through an activation function, and CNN [2][17] is no different.

3.2.2 Stride and Padding

Stride is a component of convolutional neural networks, tuned for the compression of images and video data. Stride is a parameter of the neural network filters that combines the amount of movement over the image or video. For example, if a neural network's stride is set to 1, the filter will move one pixel, or unit, at a time. The size of the filter affects the encoded output volume; therefore, stride is often set to a whole integer, rather than a fraction or decimal. Stride is the number of pixels shifts over the input matrix. Figure 3.8 describes the sample of left image with stride = 0, middle image with stride = 1 and right image with stride = 2 filtering.

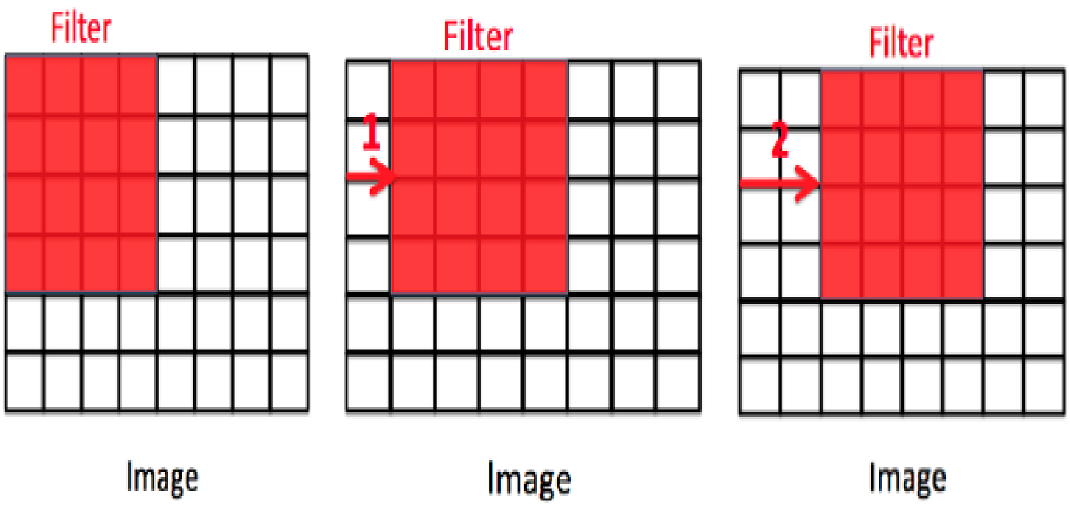


Figure 3.8 Sample of left Image: stride = 0, middle Image: stride = 1 and right Image: stride = 2

There are two problems appear with convolution described as followed:

1. After convolution operation, like the system have in example six by six down to four by four in image classification task there are multiple convolution layers accordingly after multiple convolution operation, the methodology original image will really get small but the system does not want the image to shrink every time.
2. When kernel operates through original images, it touches the edge of the image a smaller number of times and touches the middle of the image larger number of times and it overlaps also in the middle.

Therefore, in order to solve these two issues, a new concept is introduced called padding. Padding preserves the size of the original image. Since if a n x n matrix convolved with an f x f matrix width padding p. Figure 3.9 displays the sample of padding image convolved with 2x2 kernels.

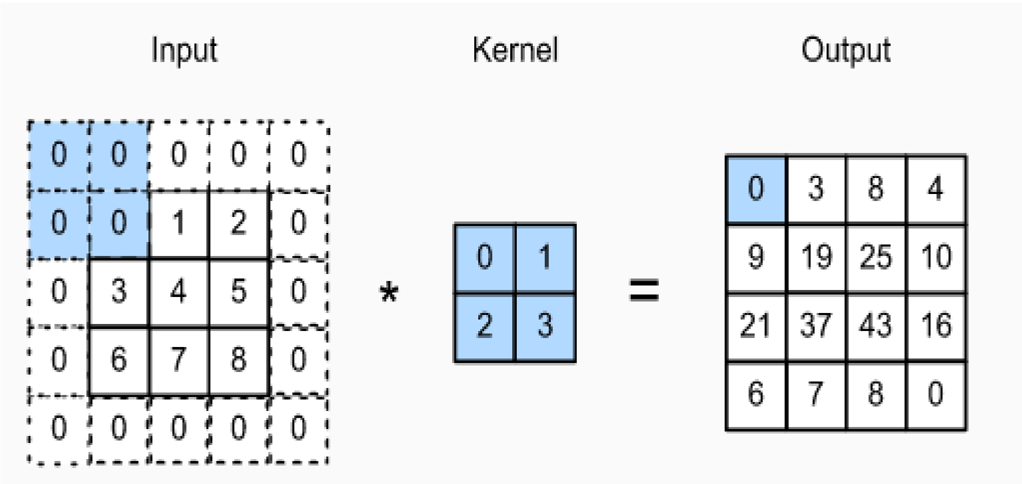


Figure 3.9 Sample of padding image convolved with 2x2 kernels

3.3 Pooling Layer

Pooling layers defined down sampling, conducts dimensionality reduction, reducing the number of parameters in the input. Similar to the convolutional layer, the pooling operation sweeps a filter across the entire input, however various is that this filter does not have any weights. The kernel applies an aggregation function to the values within the receptive field, populating the output array. The two types of pooling layer in CNN are (1) Max pooling layer and (2) Average pooling layer. The objective of the pooling layers is to decrease the dimensions of the hidden layer by adding or combining the outputs of neuron clusters at the previous layer to single neuron in the next layer.

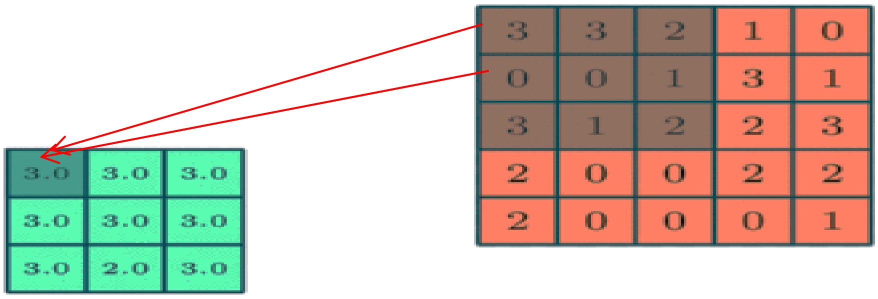


Figure 3.10 Pooling layer of 3x3 pooling over 5x5 convolved feature

The proposed system of the low-light image enhancement applied the max pooling layer operation of convolution feature. Pooling layer of 3x3 pooling over 5x5 convolved feature and max pooling are shown in Figure 3.10 and 3.11.

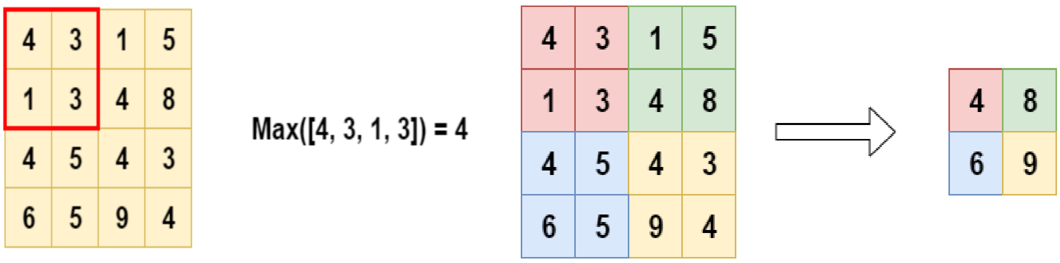


Figure 3.11 Sample of Max pooling

Average pooling is various from Max pooling in the sense that it retains much information about not important elements of a block, or pool. Where Max pooling simply throws them away by picking the maximum value, Average pooling blends them in. This can be useful in a variety of situations, where such information is useful. Figure 3.12 is sample of average pooling.

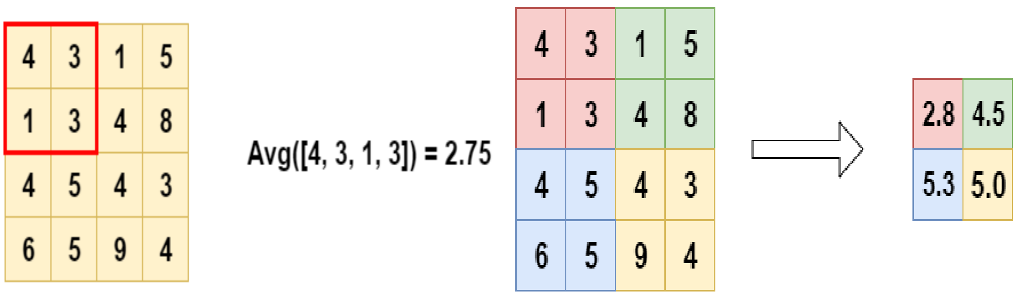


Figure 3.12 Sample of Average pooling

3.4 Fully-connected Layer

The fully-connected layer is image classification in the convolutional neural network based on the features detached in the previous layers. Fully-connected defines that all the inputs or nodes from one layer are joined or connected to every activation node or unit of the next layer. All the layers in the convolutional neural network are not the fully-connected joining because it would result output in an unneeded or uselessly dense network. It would improve losses and affect the quality of output and it would be mathematically expensive.

Regularly, this layer is located at the final or end of each convolutional neural network architecture. In this layer, each neuron is connected to all neurons of the previous layer. It is applied like the convolutional neural network classifier. It came after the basic method of the convolutional multiple layer perceptron neural network, like it is a kind of feed forward artificial neural network. The input of the fully-connected layer connects from the last pooling layer or convolutional layer. This input is in the form of a vector, which is constructed from the features maps after flattening. The output of the fully-connected layer also defined as the final convolutional neural network output. Figure 3.13 is fully-connected layer, the final layer of CNN to classify.

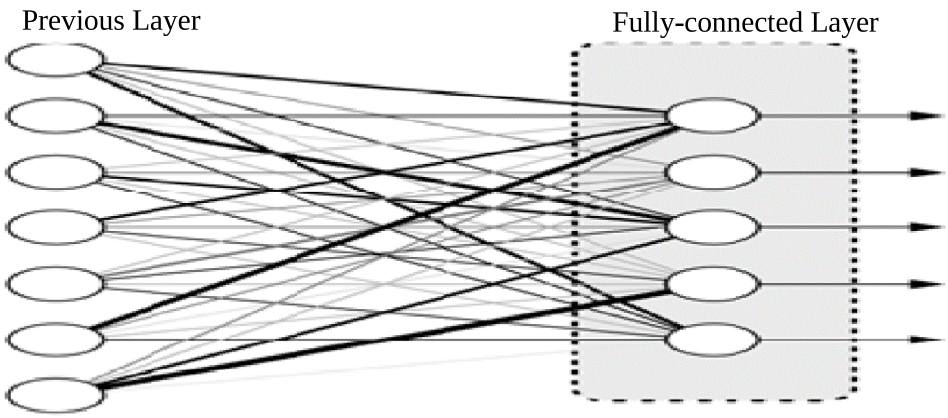


Figure 3.13 Sample of Fully-connected Layer

Input neutrons are the vector or scalar value of feature maps. Most of the hidden layer use non-linear activation function such ReLU function, hyperbolic tangent function and logistic function. The main function of Fully-connected layer is to calculate for classifying. SoftMax activation function is more widely used than other activation functions. SoftMax Activation Function Equation:

$$f(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}} \quad (3.2)$$

In this system, Sigmoid Activation Function Equation:

$$S(x) = \frac{1}{1+e^{-x}} \tag{3.3}$$

3.5 Residual Networks (ResNets) of Deep Learning

Residual Networks (ResNet) [7][8][11][17] proposed in 2015 by researchers at Microsoft research, suggested a new architecture called Residual Network. In order to solve the problem of the exploding gradient, this architecture introduced the concept called Residual blocks. In this network, the system uses a technique called skip connections. The skip connection joins or connects the activations of layer to further layers by skipping some layers in between. This forms a residual block. ResNets are made by stacking these residual blocks together. This network is alternately of layers learning the underlying mapping, the system permits the network to fit or recognize the residual mapping. Therefore, alternately of state H(x), initial mapping, let the network recognize:

$$F(x) = H(x) - x \quad \text{which gives } H(x) = F(x) + x \tag{3.4}$$

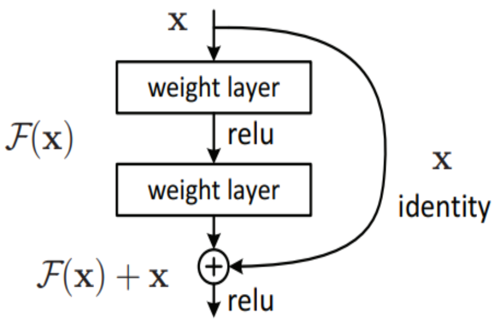


Figure 3.14 Residual learning building block of skip (shortcut) connection

In the Figure 3.14 is the most important thing to learn from this article. For developers looking to quickly implement this and test it out.

$$y = F(x, \{W_i\}) + W_s x \tag{3.5}$$

The architecture the system used the skip connections for testing followed two heuristics inspired from the VGG network [36].

1. If the system output feature maps have the same resolution e.g., 32x32 → 32x32, then the filter map depth remains the same.
2. If the system output feature map size is halved e.g., 32x32 → 16x16, then the filter map depth is doubled.

Overall, the design of a 34-layer residual network is shown in Figure 3.15.

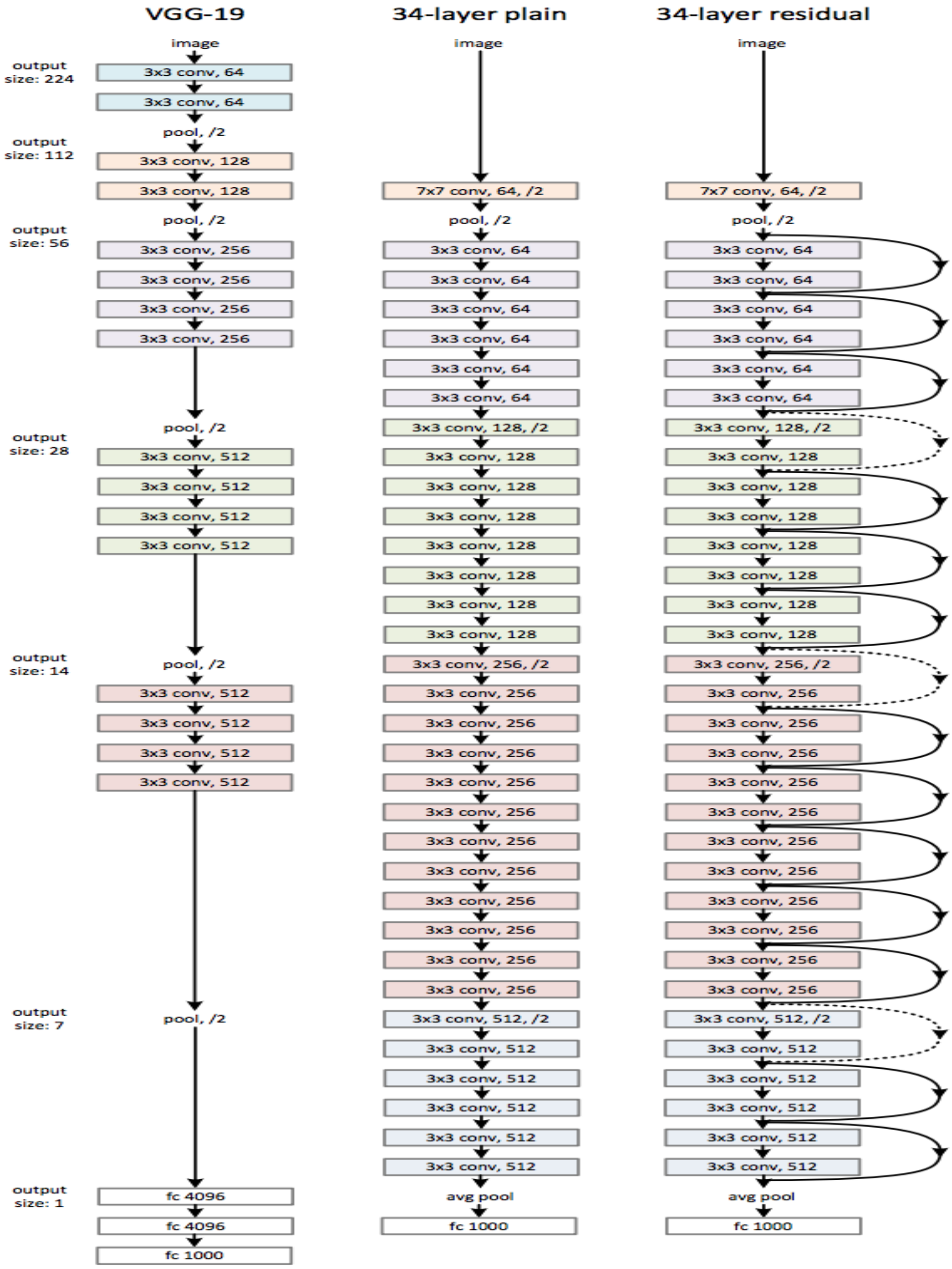


Figure 3.15 Design of 34-layer residual network the dotted skip connections

3.5.1 Residual Network (ResNet) Architecture

A residual neural network (ResNet) [7][8][11][17] is an artificial neural network (ANN). It is control gated variant of the HighwayNet, the first working deep feedforward neural with hundreds of layers, much deeper than previous neural networks. Skip connections used to jump over some layers. Typical ResNet models are implemented with double-or triple-layer skips that consist of nonlinearities (ReLU) and batch in between. Figure 3.16 is skip connections of ResNet Architecture.

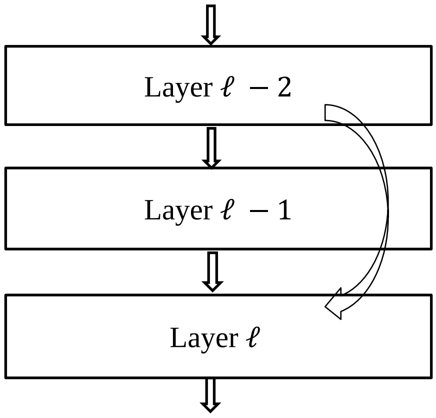


Figure 3.16 Skip connections of ResNet Architecture

ResNet is a deep learning model. ResNet is used and effective deep learning models to date. ResNets are made up of what is a residual block. This is construct on the concept of "skip-connections” of ResNet architecture and uses many of batch-normalization to permit it train hundreds of layers successfully besides offering speed over time or period. Figure 3.17 shows the system of ResNet Enhancement block.

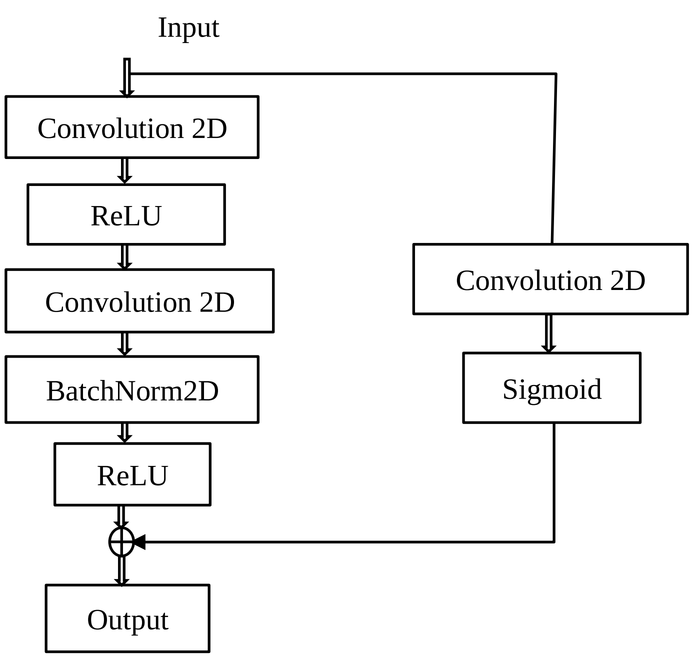


Figure 3.17 System of ResNet Enhancement Block

In this system, ResNet type of CNN model consists of 4 stages each with Convolution and Identity Block. Each Convolution block has 3 convolution layers and each Identity block has also 3 convolution layers are shown in Figure 3.18 and 3.19.

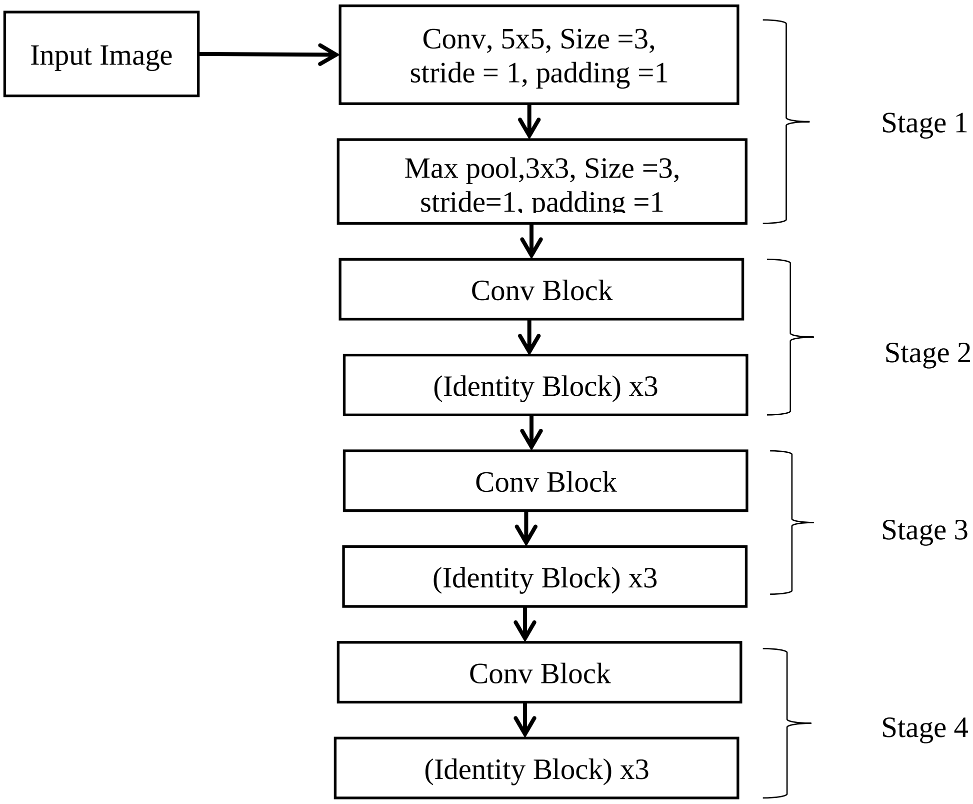


Figure 3.18 System of ResNet Enhancement Network consists of stages each with Convolution and Identity Block

3.6 Self-Calibrated Illumination

Self-Calibrated Illumination is an attempt to calibrate camera by finding intrinsic parameters with a sequence of images. Calibration of an image is a pixel into real width or height conversion factor that is the system calibration of factor, pixels/cm, that permits image scaling to metric units. The system purposes at referring a component of module to make results output of each stage convergent to the equivalent one state. The system defines the input of each stage finishes from the previous stage and the input of the first stage is certainly describe the low-light observation. The system can extension the input of each stage and the low-light observation that is the input of the first stage to not immediately explore the convergence behavior among each stage. The system of an Illumination parameters can convert the overall magnitude of light intensity reflected back from an object. The system introduces a self-calibrated map s and add it to the low-light observation to present the difference between the input in each stage and the first stage. Figure 3.19 is self-calibrated illumination (SCI) frame work. When the conversion stage is in the system of t-th stage (t >= 1) can be written as follow:

$$\mathbf{F}(x^t) \rightarrow \mathbf{F}(\mathbf{G}(x^t)) \quad (3.6)$$

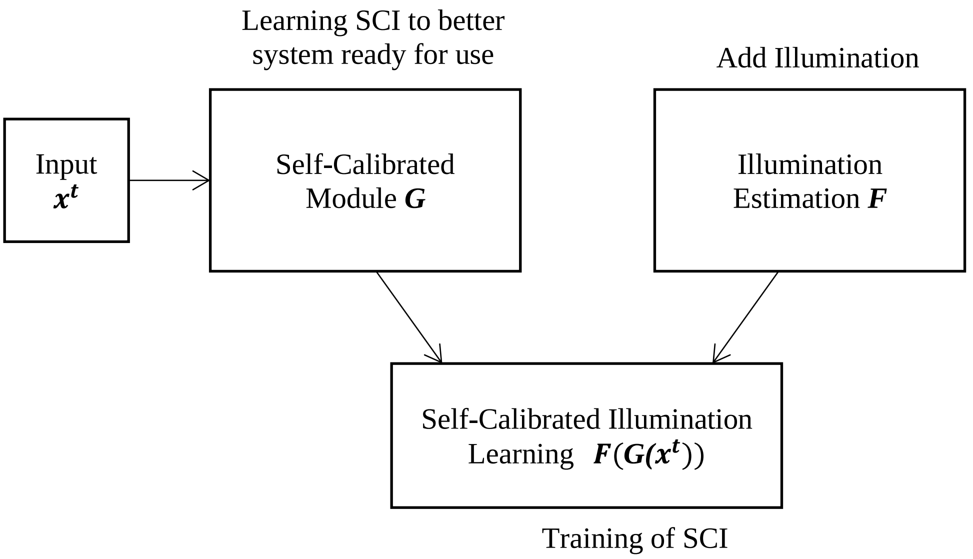


Figure 3.19 Self-Calibrated Illumination (SCI) framework

Self-Calibrated Illumination of Block, System of Self-Calibrated Network consists of 10 stages each with Convolution and Identity Block with training of system stages are shown in Figure 3.20 and 3.21.

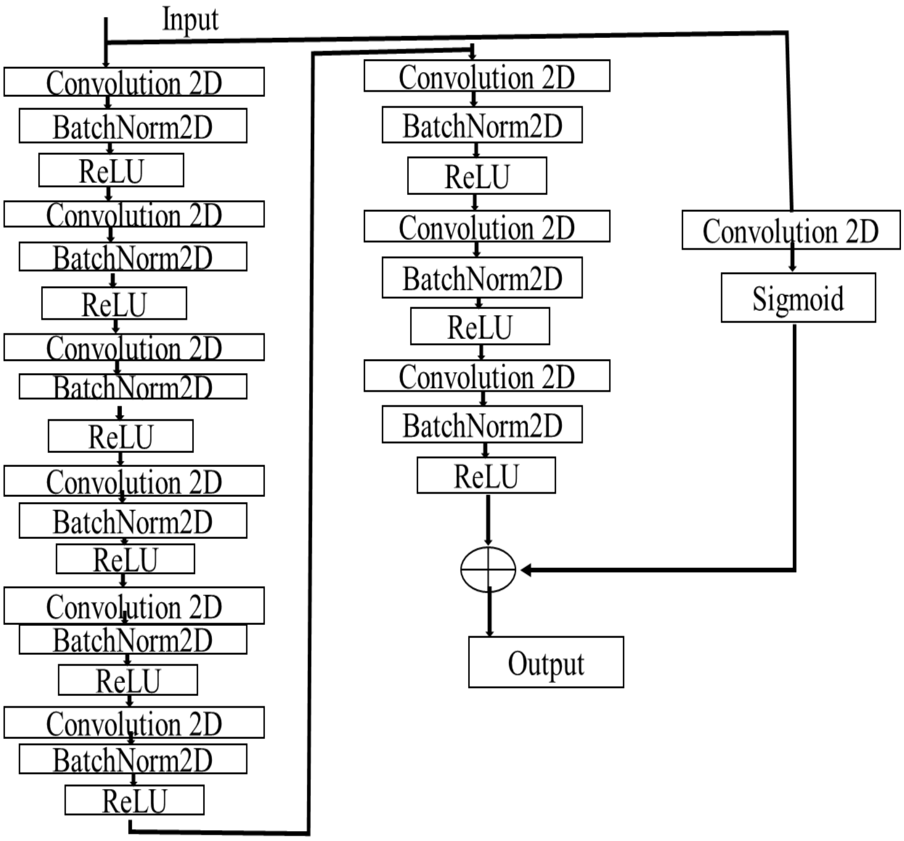


Figure 3.20 System of Self-Calibrated Block

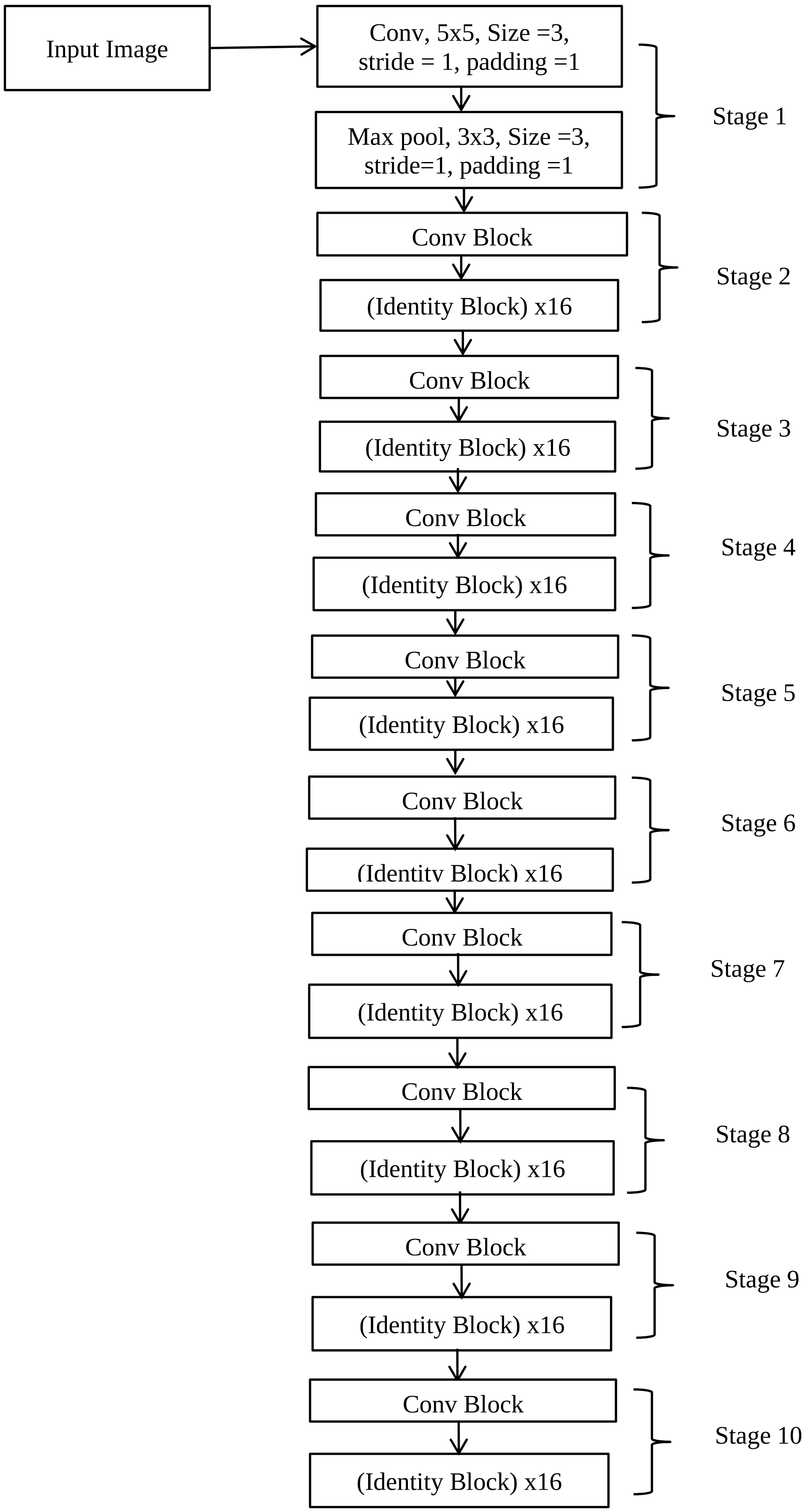


Figure 3.21 System of Self-Calibrated Network consists of 10 stages each with Convolution and Identity Block

CHAPTER 4

SYSTEM IMPLEMENTATION AND EXPERIMENTAL RESULTS

In this chapter, the proposed system is described for low-light image enhancement based on CNN by adding self-calibrated illumination network. The preparation of dataset, the implementation of the system and experimental results are explained.

4.1 The Proposed System

The proposed system flow of low-light image enhancement with CNN is shown in Figure 4.1.

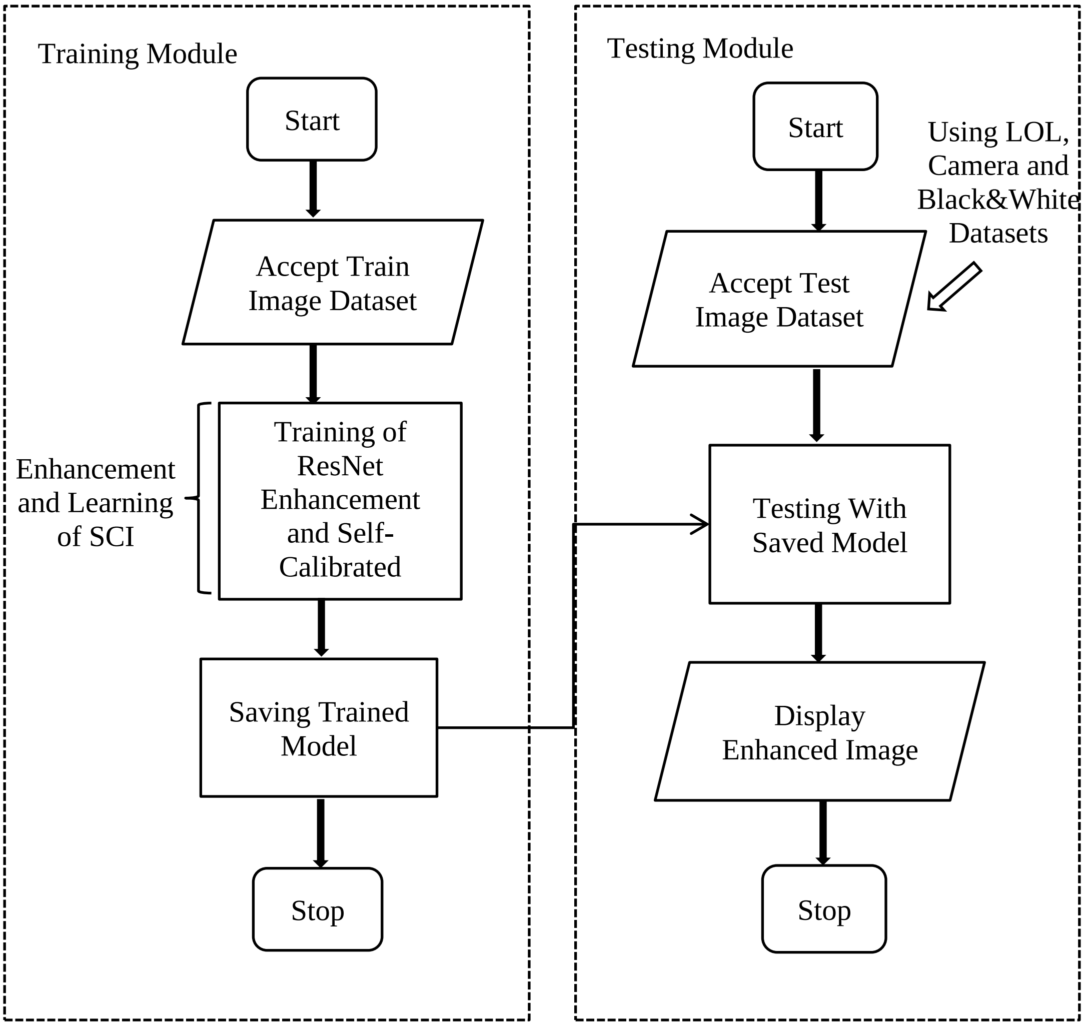


Figure 4.1 System Flow Chart

The proposed system of low-light image enhancement is based on CNN by adding self-calibrated illumination system by using convolution neural network (CNN). There has been many CNN based on techniques. Among these techniques, this system used the ResNet [11] enhancement network and learning of self-calibrated illumination (SCI) network. At the first of the system, the user necessary of applications should download and install. Then, the user chooses preparing the training and testing module. To classify the classification of low-light image, this system performs training process as CNN based on ResNet enhancement network and learning of self-calibrated illumination (SCI) network. The networks are trained using the low-light dataset. After the training step, this system saves trained model. Python programming language is used for developing the model. And, in the testing step, the user tests with the training model. Finally, this system displays the high resolution of low-light image enhancement.

4.2 Requirements of the System Processing

The following steps are requirements of the system processing in low-light image enhancement.

- Downloading the Anaconda Prompt (Anaconda3) application
- Installation application and necessary of python and environment
- Downloading LOL dataset for training and testing
- Move the LOL dataset to Dataset folder in the SCI-main
- Applying the construct of ResNet Enhancement Network and SCI Network to enhance as model of training
- Saving the trained model .py file to Model folder in the SCI-main
- Testing the low-light image using saved training model
- Displaying the output high resolution of low-light image enhancement

This system used python version of 3.9.12 and CPU module. CPU cards and modules are computer boards that consist of the Central Processing Unit (CPU). A CPU module is a connectorized CPU with the external logic needed to permit it to function. Some languages, like java, allow us to run code in parallel on multiple CPU's. Python, however, is single-threaded on a single CPU by design. And, a single source tree of Compute Unified Device Architecture (CUDA) code provides applications that run completely on convolutional x86 processors, completely on Graphics Processing Unit

(GPU) hardware like hybrid applications that use all the CPU and GPU devices in a system to display maximal performance.

4.3 Low-light Image Dataset

These system uses low-light image dataset. Low-light images especially consist of low-light regions and images captured low-light conditions often display characteristics like low illumination, low contrast, and low gray range. LOL dataset includes dimensions (width 1080 pixels x height 720pixels) PNG file type low-light images. The LOL dataset contains 6,000 low light images captured such as parks, bridges, humans, streets, faces, things, the nighttime, and so on. LOL dataset of 3,945 is used for training and 2,150 is used for testing. This system is used for testing with standard LOL dataset 2,150, Camera dataset 386 and Black&White dataset 150. Figure 4.2 shows the sample images of LOL dataset.

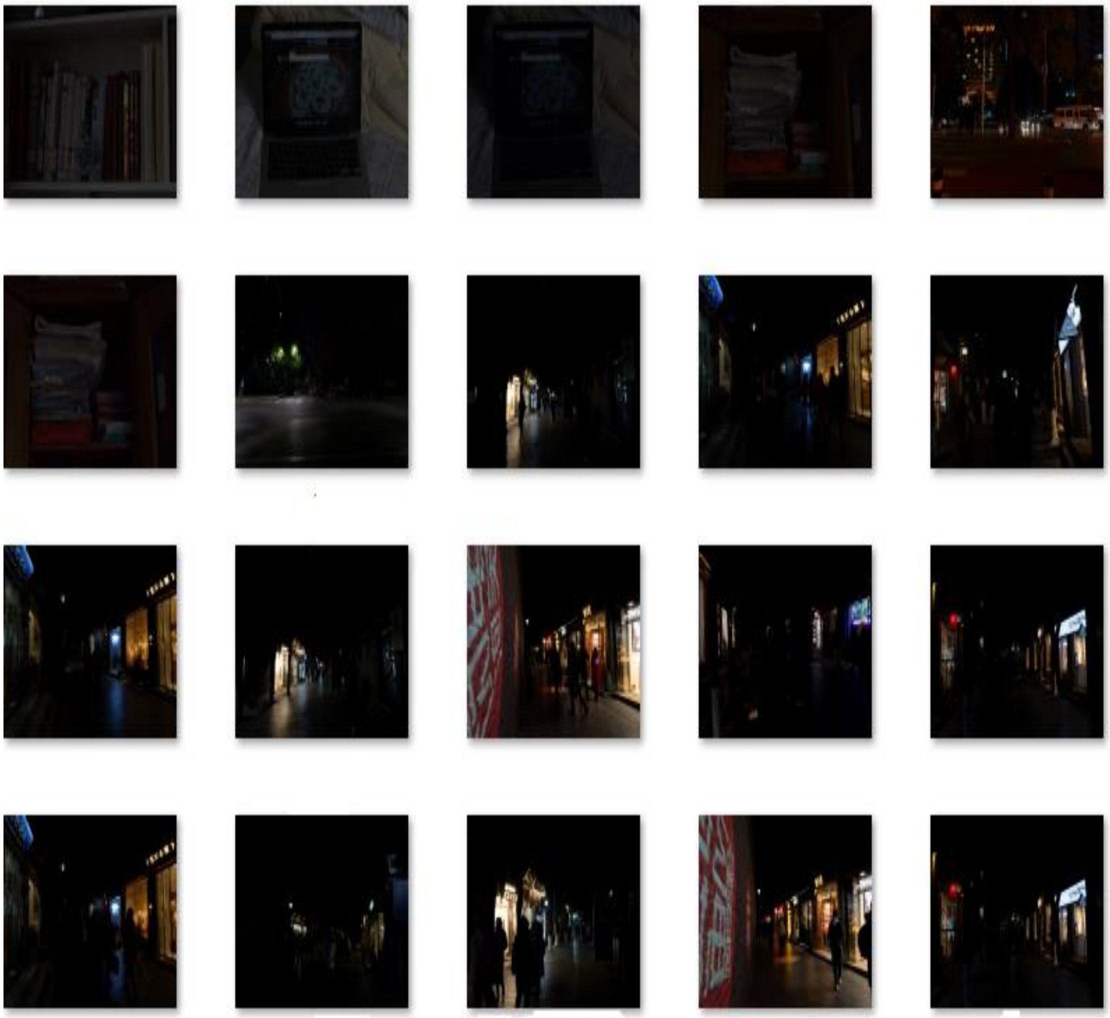


Figure 4.2 Sample Images from LOL Dataset

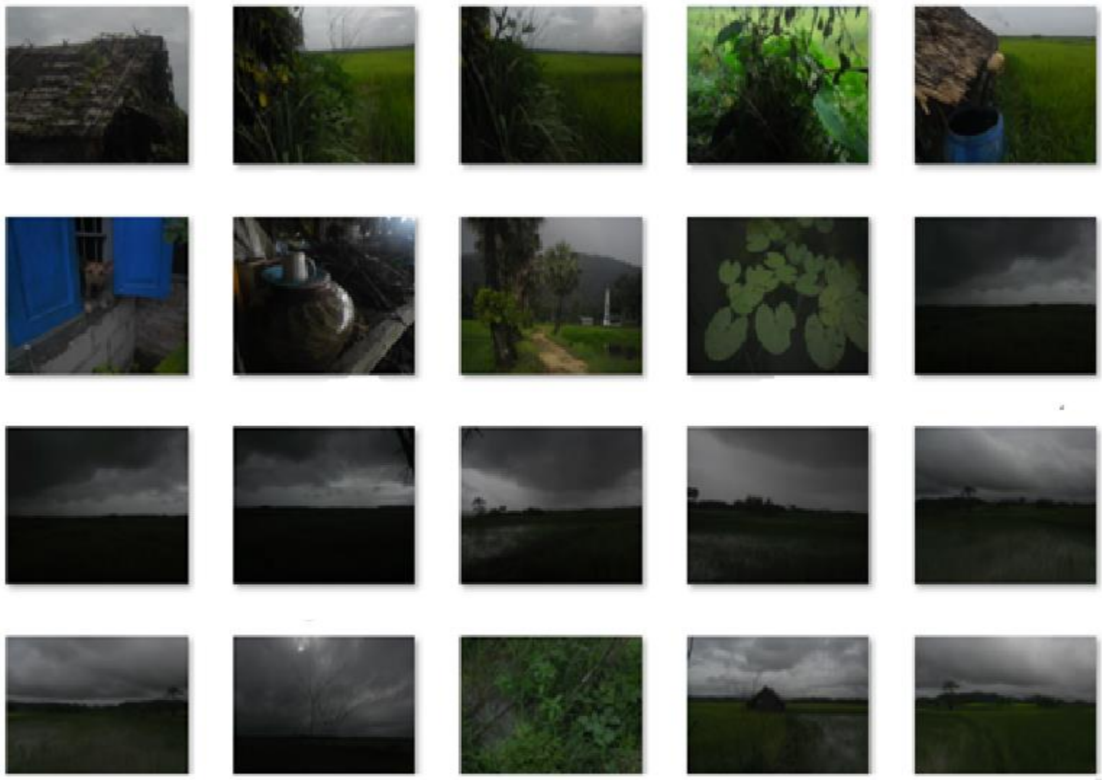


Figure 4.3 Sample Captured Images in Low Illumination

Figure 4.3 shows the sample images of camera dataset. In addition, balck&white images are used for testing. Some images of black&white dataset are shown in Figure 4.4.

Figure 4.4 Sample Images of Black&White dataset

4.4 Implementation of the System

The low-light image enhancement system is implemented with using Python programming language. In the proposed system process, this system applied the ADAM optimizer, the name is derived from adaptive movement estimation with the parameters β_1 = 0.0666, β_2 = 0.00009999 and ϵ = 10^{-8}. ADAM optimization is an addition to stochastic low gradient and can be used in place of classical stochastic gradient slope to update network weights better efficiently. ADAM is a replacement optimization algorithm for stochastic gradient slope for training deep learning models. ADAM connects the best properties of the Adaptive Gradient (AdaGrad) and Root Mean Square Propagation (RMSProp) algorithms to contribute an optimization algorithm that continue manage poor gradients on noisy problems. The python stream processing for people size was determined to 8. The system learning rate was initialized to 10^{-4}. The system training epoch number was set to 1000. The system adopts 3 convolutions + ReLU with 3 channels just as this system in action setting for H_θ in all experiments in keeping with conclusion.

The operations applied network-based methods also be stable and unable be restored immediately since these operations are acquired less than the experiments. Opportunely, this system suggested algorithm arrives amazing flexibility on various highly complete, stable complete setting for H_θ. $H\theta$ is the output of the hypothesis function.

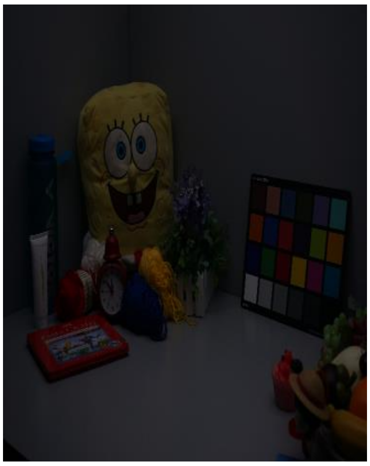
Table 4.1 Different settings for H_θ of blocks and channels on LOL dataset testing

Setting for H_θ		**Metric**	**Efficiency**	
Blocks	Channels	PSNR	Delta	Time (s)
1	3-3	54.9074	0.1000	0.0600
2	3-3-3	54.8809	0.1000	0.0680
3	3-3-3-3	54.7943	0.1000	0.0750
3	3-8-8-3	54.5779	0.0090	0.0870
3	3-16-16-3	54.5215	0.0090	0.0950

Table 4.1 describes the different values of setting for Blocks and Channels, metric performance of PSNR, and efficiency of Delta and Time(s) on LOL dataset testing. When using Block 1, Channel 3-3, and Delta 0.1000, the metric performance

of PSNR 54.9074 at Time(s) 0.0600. When using Block 2, Channel 3-3-3, and Delta 0.1000, the metric performance of PSNR 54.8809 at Time(s) 0.0680. When using Block 3, Channel 3-3-3-3, and Delta 0.1000, the metric performance of PSNR 54.7943 at Time(s) 0.0750. When using Block 3, Channel 3-8-8-3, and Delta 0.0090, the metric performance of PSNR 54.5779 at Time(s) 0.0870. When using Block 3, Channel 3-16-16-3, and Delta 0.0090, the quality of PSNR 54.5215 at Time(s) 0.0950. This system acquired a stable performance between various settings.

This system provides the comparisons different causes in Figure 4.5, 4.6 and 4.7. This experiment also confirms the success and exactness of this system designed SCI.



Input

Layer 1, Channel 3, delta 0.1,
Enhance = ill+input+delta,
Calibrate ill+input+delta

Layer 2, Channel 3, delta 0.1,
Enhance = ill-input+delta,
Calibrate ill+input+delta

Layer3, Channel 3, delta 0.1,
Enhance = ill-input-delta,
Calibrate ill-input+delta

Layer 3, Channel 8, delta 0.009,
Enhance = ill-input+delta,
Calibrate ill-input-delta

Layer 3, Channel 16, delta 0.009,
Enhance = ill-input-delta,
Calibrate ill-input-delta

Figure 4.5 Comparison Results of LOL datasets based on different parameters

Input

Layer 1, Channel 3, delta 0.1,
Enhance = ill+input+delta,
Calibrate ill+input+delta

Layer 2, Channel 3, delta 0.1,
Enhance = ill-input-delta,
Calibrate ill+input+delta

Layer3, Channel 3, delta 0.09,
Enhance = ill-input+delta,
Calibrate ill+input+delta

Layer 3, Channel 8, delta 0.09,
Enhance = ill-input-delta,
Calibrate ill+input-delta

Layer 3, Channel 16, delta 0.03,
Enhance = ill-input-delta,
Calibrate ill-input-delta

Figure 4.6 Comparison Results of Images Captures by Camera based on different parameters

Input

Layer 1, Channel 3, delta 0.1,
Enhance = ill+input+delta,
Calibrate ill+input+delta

Layer 2, Channel 3, delta 0.009,
Enhance = ill-input+delta,
Calibrate ill+input+delta

Layer3, Channel 3, delta 0.05,
Enhance = ill-input-delta,
Calibrate ill-input-delta

Layer 3, Channel 8, delta 0.09,
Enhance = ill-input-delta,
Calibrate ill-input-delta

Layer 3, Channel 16, delta 01,
Enhance = ill-input-delta,
Calibrate ill-input-delta

Figure 4.7 Comparison Results of Images on the Black&White dataset based on different parameters

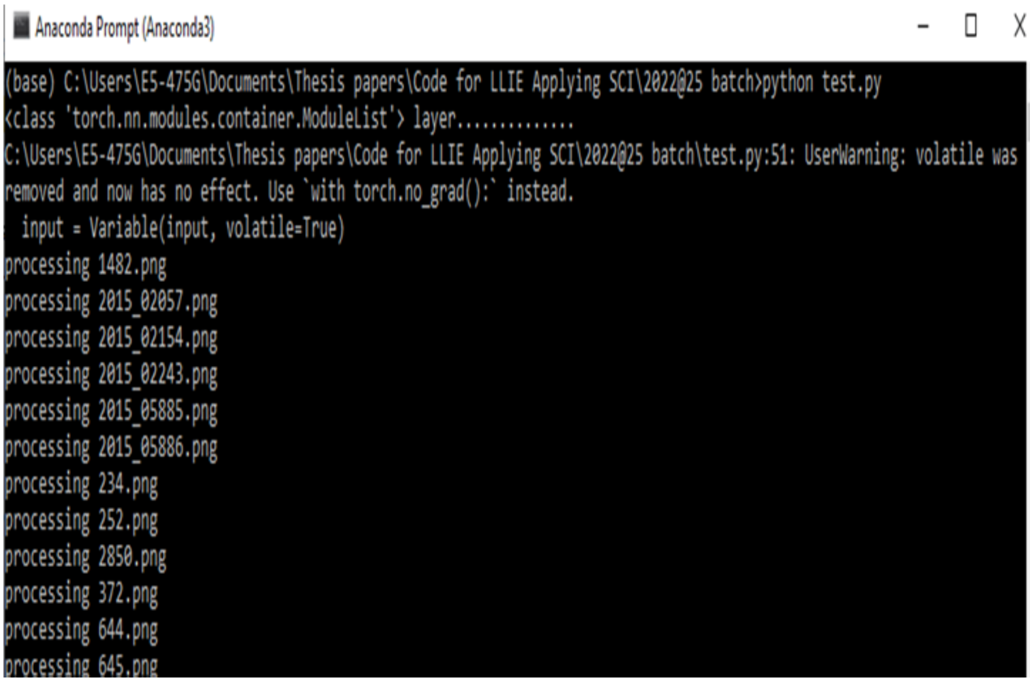


Figure 4.8 Sample Testing Process of LOL images using ResNet and SCI Network

In Figure 4.8, the testing of this implemented system run command with python test.py on the Anaconda Prompt (Anaconda3) used sample 12 images. Figure 4.9 original low-light images to display the output enhancement images using ResNet Enhance Network and Self-Calibrated Enhancement Network.

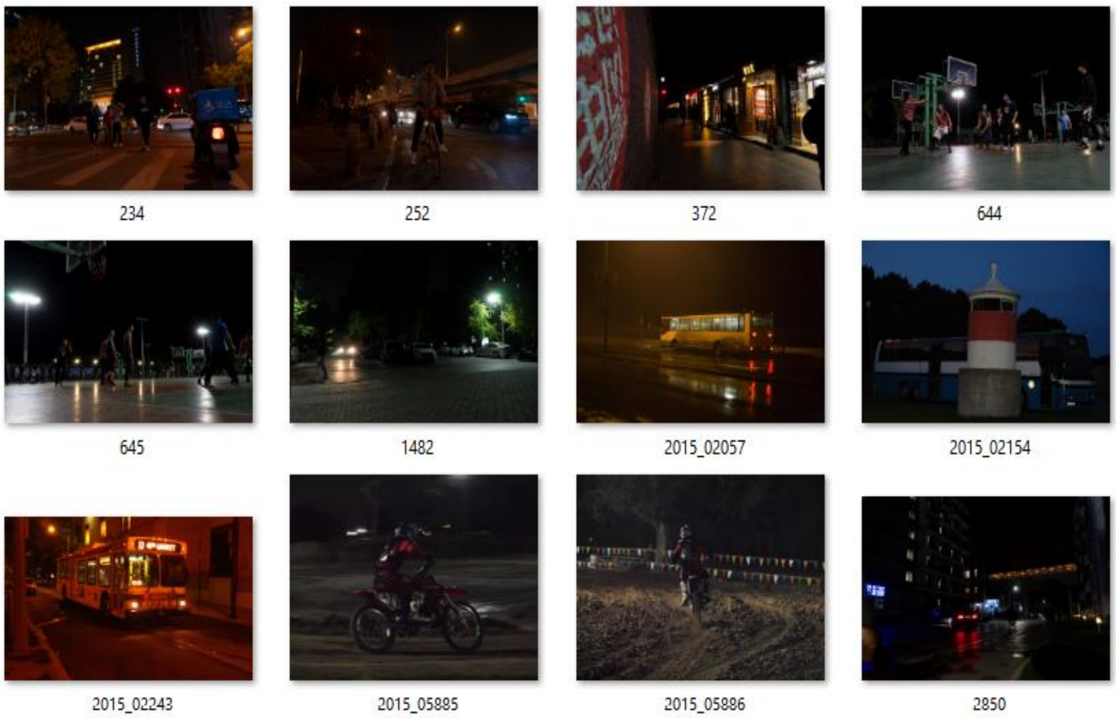


Figure 4.9 Original low-light images before using ResNet Enhancement Network

Figure 4.10 Original low-light images after more delta and illumination values using ResNet Enhancement Network

In the Figure 4.10 and 4.11 are displaying the output images from processing run python test.py in Figure 4.8 using ResNet Enhancement Network with original low-light images after more and less values of delta and illumination.

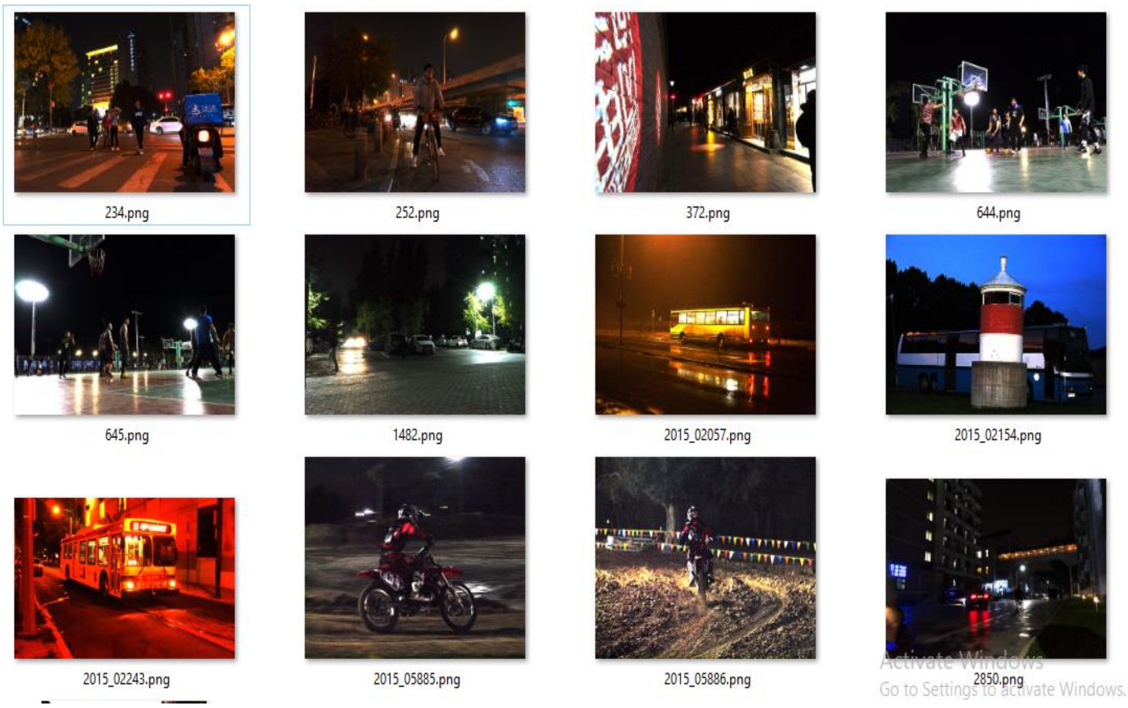


Figure 4.11 Original low-light images after less delta and illumination values using ResNet Enhancement Network

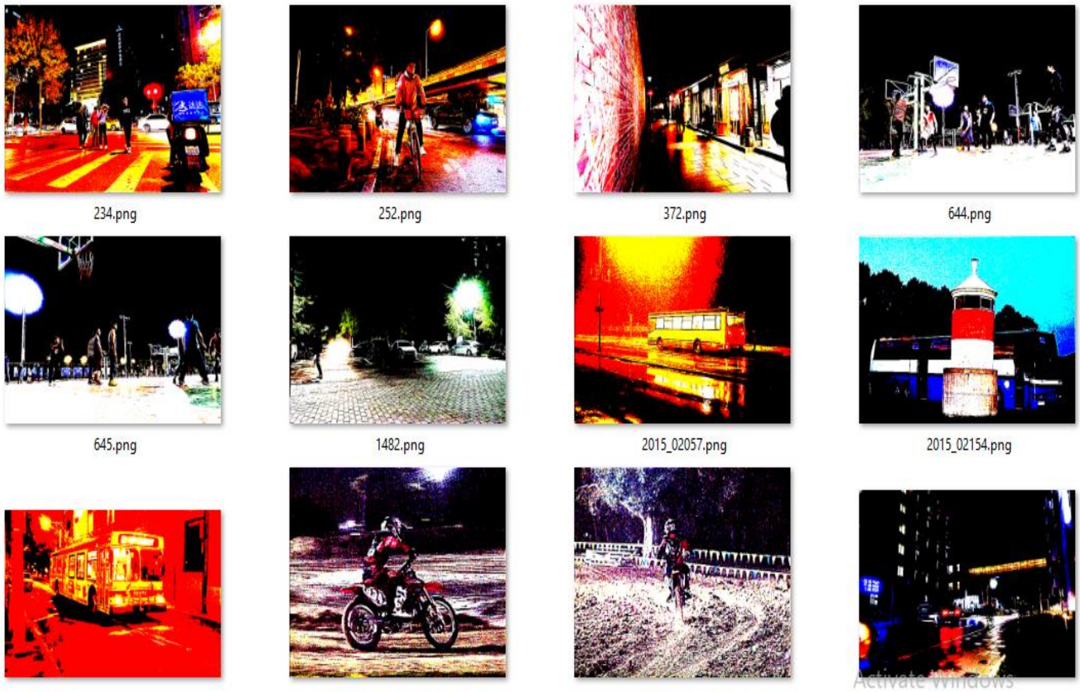


Figure 4.12 Original low-light images after more delta and illumination values using Self-Calibrated Enhancement Network

In the Figure 4.12 and 4.13 are displaying the output images from processing run python test.py in Figure 4.8 using ResNet Enhancement Network with original low-light images after more and less values of delta and illumination.

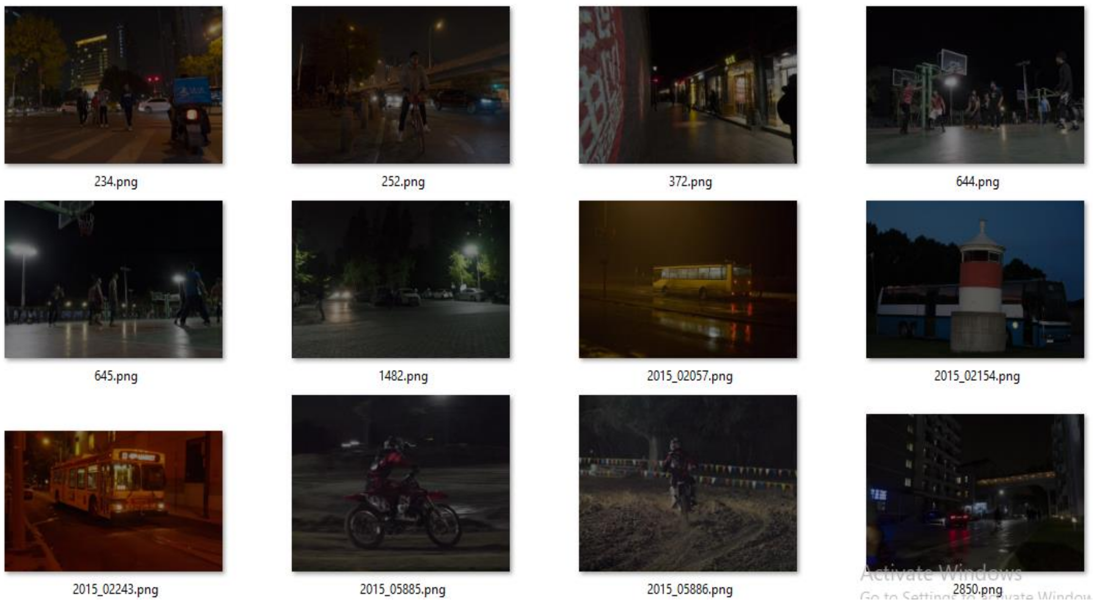


Figure 4.13 Original low-light images after less delta and illumination values using Self-Calibrated Enhancement Network

4.5 Experimental Results

In this system, the experimental results are tested with enhance model using different three datasets of LOL, real camera, and black & white on the Anaconda Prompt (Anaconda3). The system output of high-resolution image display using Viewer Image folder of python code run command with python main.py on the Anaconda Prompt (Anaconda3) and comparison of original low-light image and enhanced the high-resolution output image.

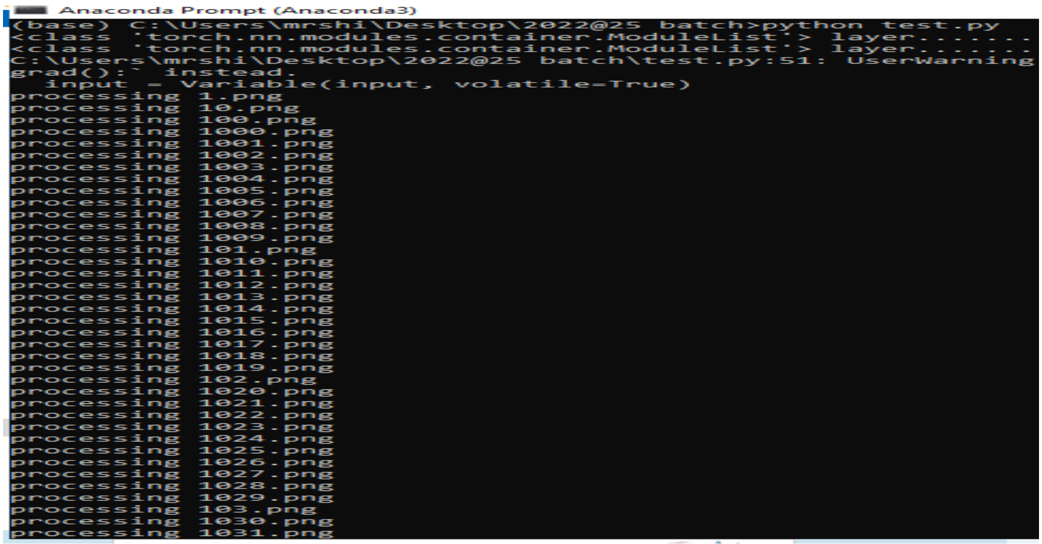


Figure 4.14 Testing Process of LOL Images

Figure 4.14 shows the processing of LOL testing images run with command python test.py on the Anaconda Prompt (Anaconda3) using enhancement model to display output images, testing process of LOL images.

In addition, the low-light image enhancement [4][12][15] system develops display the low-light image to convert the high resolution of result enhancement image using the python tkinter GUI code. And, this system will be constructed learning the necessary setting format repairing and combination of new code. The GUI of tkinter window shows the low-light image. This system will be used displaying as the three types of low-light image dataset. Moreover, this system will be tested showing the different between original upload low-light image and changing the result output image. The processing of low-light image enhancement will be converted the final result output image using the enhance model of steps and adding the illumination in the original input image. This system is using the enhance model of steps and adding the illumination in the original input image. Figure 4.15 shows the main page of the proposed system.

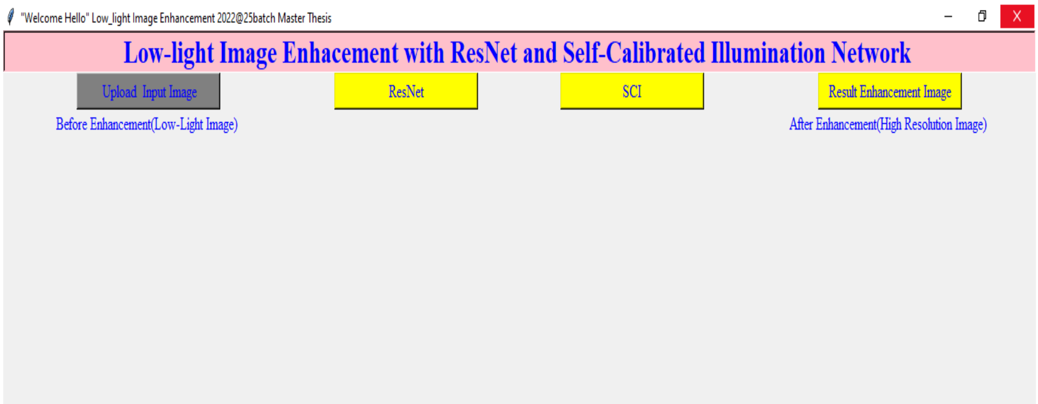


Figure 4.15 The main page of the proposed system

Figure 4.16 displays the output image after enhancement using ResNet Enhancement on LOL dataset when the button of “ResNet” clicked.

Figure 4.16 Using ResNet Enhancement on LOL dataset

Figure 4.17 shows after enhancement using SCI Enhancement on LOL dataset when the button of “SCI” clicked.

Figure 4.17 Using SCI Enhancement on LOL dataset

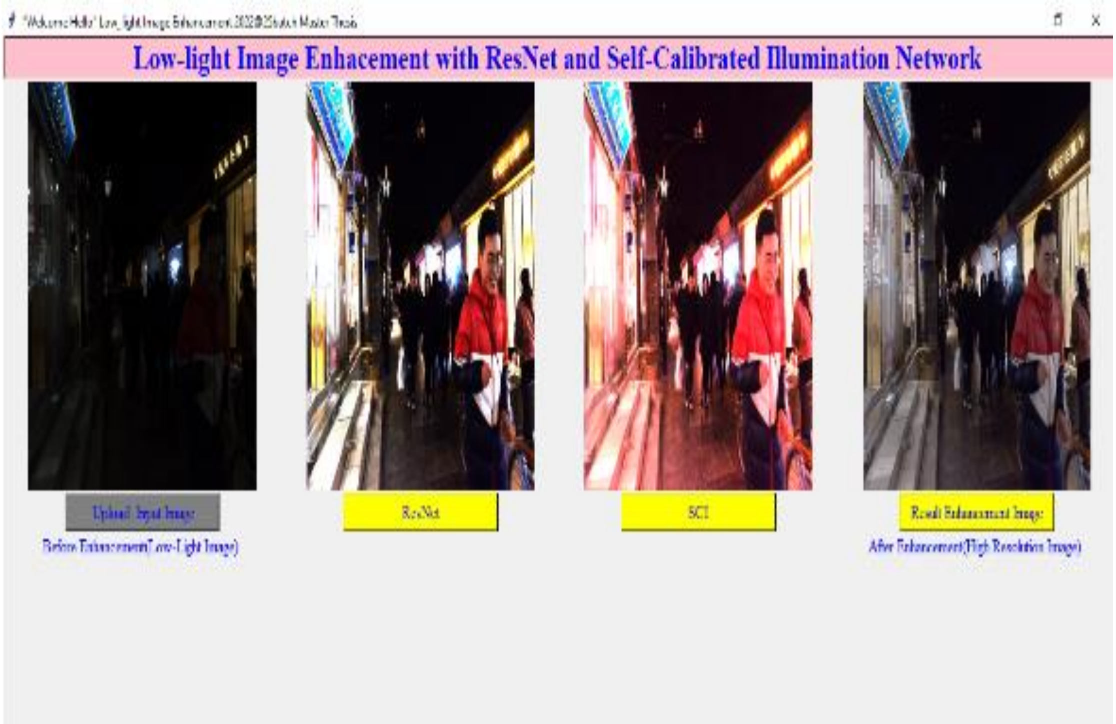


Figure 4.18 After enhancement processing of ResNet and SCI using LOL dataset

Figure 4.18 describes the final result changing processing after enhancement processing of ResNet and SCI using LOL dataset on the main window when above the label "After Enhancement (High Resolution Image)" of the button "Result Enhancement Image" clicked.

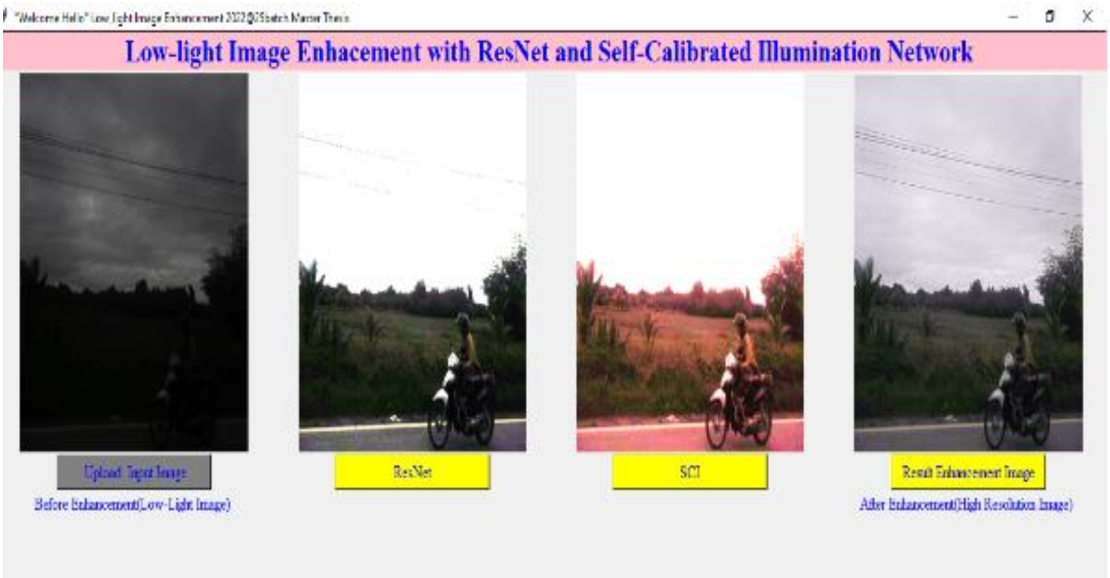


Figure 4.19 After enhancement processing of ResNet and SCI using Camera dataset

Figure 4.19 shows the final result changing processing after enhancement processing of ResNet and SCI using Camera dataset on the main window when above the label "After Enhancement (High Resolution Image)" of the button "Result Enhancement Image" clicked.

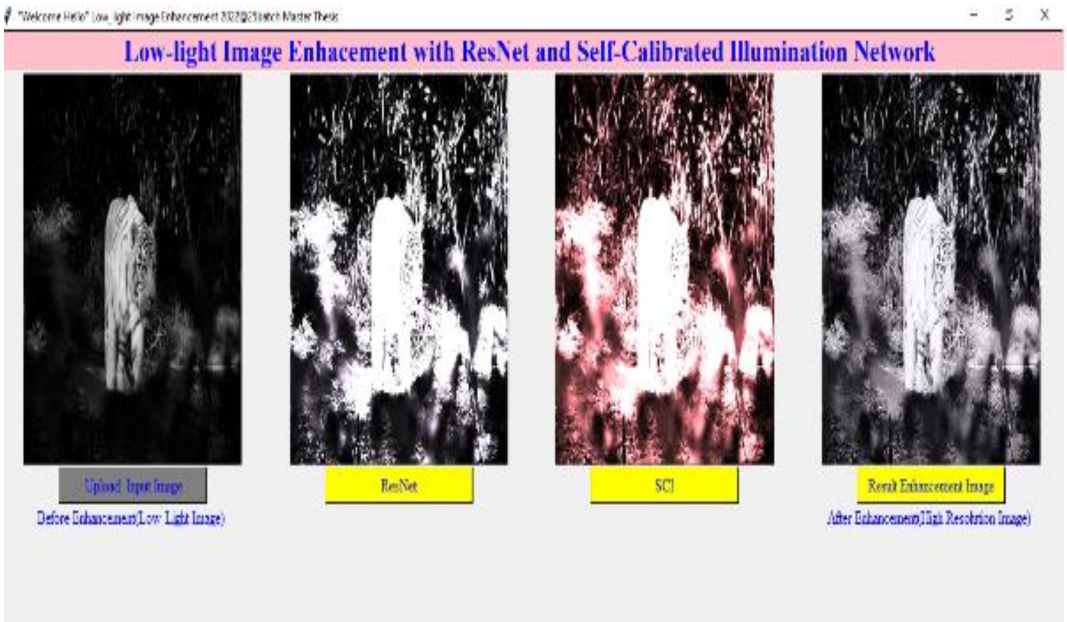


Figure 4.20 After enhancement processing of ResNet and SCI using Black&White dataset

Figure 4.20 displays the final result changing processing after enhancement processing of ResNet and SCI using Black&White dataset on the main window when above the label "After Enhancement (High Resolution Image)" of the button "Result Enhancement Image" clicked.

4.5.1 Performance Evaluation

Performance evaluation of SCI achieved competitive performance as shown in following Table 4.2, especially in no reference metrics. In addition, computational efficiency of the system, further, the system reported the model dimensions, flat, and running period (CPU or GPU-second) in Table 4.1. Obviously, this proposed system of SCI is the best lightweight compared with other networks. Table 4.2 describes the performance evaluation results on the LOL dataset. Like metrics are PSNR and SSIM. Unsupervised learning methods are ResNet, SCI and ResNet + SCI Enhancement. When using ResNet, the value of PSNR is 52.4207, SSIM is 0.9328, Accuracy is 83% and Time is 0.8317. When using SCI, the value of PSNR is 51.6648, SSIM is 0.9216, Accuracy is 81% and Time is 0.8425. When using ResNet + SCI, the value of PSNR is 54.7943, SSIM is 0.9547, Accuracy is 85% and Time is 0.9714. The good value of PSNR is 30dB to 60dB. Therefore, the system of PSNR value is good. The SSIM values range between 0 to 1. The good value of SSIM is 0.8 to 1. The system of SSIM value is good. Accuracy better value is over 90%. Accuracy good value is between 70% and 90%. Thus, the system of Accuracy value is good.

Table 4.2 Performance Evaluation results in terms of PSNR, SSIM and Accuracy on the LOL datasets

Dataset	**Metrics Accuracy% and Time**	**Unsupervised Learning Methods**		
		ResNet Enhance-ment	**SCI Enhance-ment**	**ResNet+SCI Enhance-ment**
LOL	**PSNR (dB)**	52.4207	51.6648	54.7943
	SSIM	0.9328	0.9216	0.9547
	Accuracy%	83%	81%	85%
	Time	0.8317	0.8425	0.9715

CHAPTER 5

CONCLUSION

Low-Light Image Enhancement (LLIE) defines at increasing the attention or illustratable of an image taking in an environment with low-light. The enhancement system is used to build it for comparisons different interpretation and deciding of imagery. The low-light images are enhancement by integrating CNN based ResNet model and SCI framework. The result of the enhanced image of low-light image is more effective by integrating of the ResNet and SCI. In this study, proposed integrated system is achieved to get brighter image from low-light images. It improves in value the illumination and the details of the low-light images while preserving the serenity.

In the proposed system, CNN based on ResNet architecture enhancement network and the self-calibrated illumination network system has been successfully constructed combination. Like testing of three low-light dataset are used LOL, Camera and Black&White dataset. Finally, the changing different of input low-light image compare displayed the output of high-resolution image.

5.1 Benefits of the System

The system can provide increasing the attention or illustratable of an image taking in an environment with low illumination. To build the system of computer vision algorithms fit in low-light states, use low-light image enhancement to increase the visibility of an image. The system of low-light image enhancement-based CNN automatically enhanced the original input of low-light image. And this system obtained more interpretation and visualization effects. In addition, this system provided the low-light image more viewing during the limit time of processing. Finally, this system applied self-calibrated learning framework in low-light images enhancement process to be fast, flexible, and robust brightening.

5.2 Limitations of the System

In this thesis, the system of low-light image enhancement based on CNN used python code with CPU. Some languages, like Java, allow us to run code in parallel on multiple CPU's. Python, however, is single-thread on a single CPU by design. Sophisticated python code and the applications the system build later require a solid CPU. It is the heart of the computer after all. These systems give Intel i5 and i7 processors, mostly 8th, 9th or 10th generation. Intel i9 is hardly found in laptops, it is just too expensive. The weaker i3 is not worth considering, mostly since it is not much cheaper. If these systems are applying with troch.no.grad (), this system specifically add system refuse operate gradients for the operations internal this block, therefore Loss.backward () will not work. Usually, these systems apply it for the test dataset. If these systems require to call Loss.backward () this system could not apply it. Although python has many advantages, no language is a perfect solution for all the needs. Though python has several strengths over almost all the other programming languages, it also has some disadvantages/limitations.

5.3 Further Extension

The system successfully constructed a lightweight of low-light image enhancement system. Further extension, the system future work will target on improving the generalization ability of the enhancement model and enhancement effect in extreme environments, as well as constructing a complete or full enhancement and object detection system for nighttime autonomous driving and video surveillance. This system can modify the dark face detection, night security and other detection of low-light conditions.

PUBLICATIONS

[1] Zayar Tun and Khant Kyawt Kyawt Theint, “Low-Light Image Enhancement with ResNet Architecture and Self-Calibrated Illumination Network”, University of Computer Studies, Yangon, Myanmar, 2022.

REFERENCES

[1] Ali Borji, Ming-Ming Cheng, Qibin Hou, Huaizu Jiang, and Jia Li. Salient object detection: A survey. Computational Visual Media, 5(2):117–150, 2019.1

[2] CNN:https://cs231n.github.io/convolutional-networks/

[3] Fu X, Zeng D, Huang Y, et al. "A weighted variational model for simultaneous reflectance and illumination estimation," IEEE Conference on Computer Vision and Pattern Recognition. 2016, pp. 2782-2790.

[4] Guo X, Li Y, Ling H. "LIME: Low-Light Image Enhancement via Illumination Map Estimation, vol. 26, no.2, 2017.

[5] Image Enhancement: https://www.nrcan.gc.ca/maps-tools-and-publications/satellite-imagery-and-air-photos/tutorial-fundamentals-remote-sensing/image-interpretation-analysis/image-enhancement/9389

[6] Jiwon Kim, Jung Kwon Lee and Kyoung Mu Lee, "Accurate Image Super-Resolution Using Very Deep Convolutional Networks," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1646-1654

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun: "Deep Residual Learning for Image Recognition, Dec, 2015.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Identity Mappings in Deep Residual Networks. 2016.

[9] Lim, J., Kim, J. H., Sim, J. Y., Kim, C. S., 2015. Robust contrast enhancement of noisy lowlight images: Denoising-enhancement-completion. In: Image Processing (ICIP), 2015 IEEE International Conference on. IEEE, pp. 4131–4135.

[10] Qingnan Fan, Jiaolong Yang, David Wipf, Baoquan Chen, and Xin Tong. Image smoothing via unsupervised learning. ACM Transactions on Graphics), 37(6):1–14, 2018.

[11] ResNet:https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/

[12] Shen, L., Yue, Z., Feng, F., Chen, Q., Liu, S., Ma, J., 2017. Msr-net: Low-light image enhancement using deep convolutional network. arXiv preprint arXiv: 1711.02488.

[13] Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang and Philip Torr, "Res2net: A new multi-scale backbone architecture", 2020.

[14] Wei Chen, Wenjing Wang, Wenhan Yang, and Jiaying Liu.‘‘Deep retinex decomposition for low-light enhancement’’.

[15] Yurui Ren, Zhenqiang Ying, Thomas H Li, and Ge Li.Lecarm: “Low-light image enhancement using the camera response model” , 2018.

[16] *Zhang, Wei (1988). “Shift-invariant pattern recognition neural network and its optical architecture”. Proceediings of Annual Conference of the Japan Society of Applied Physics.*

[17] Zhang K, Zuo W, Chen Y, et al. “Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising”, IEEE Transactions on Image Processing, 2017.