Mobile Application Development with Image Applications Using Xamarin

GAJJELA VENKATA SARATH KUMAR REDDY

DUPATI SURYA DEEPTHI RAJU

DECEMBER 2017

Faculty of Engineering
Department of Applied Signal Processing
Supervisor: Dr. Benny Lövström
Blekinge Institute of Technology
SE-37179 Karlskrona Sweden

Contact Information:

Author(s)
Gajjela Venkata Sarath Kumar Reddy
E-mail: vega16@student.bth.se

Dupati Surya Deepthi
E-mail: sudu16@student.bth.se

University advisor:
Dr. Benny Lövström

Examiner:
Dr. Sven Johansson

Department of Applied Signal Processing
Faculty of Engineering
Blekinge Institute of Technology
SE-37179 Karlskrona, Sweden
Internet: www.bth.se

ABSTRACT

Image enhancement improves an image appearance by increasing dominance of some features or by decreasing ambiguity between different regions of the image. Image enhancement techniques have been widely used in many applications of image processing where the subjective quality of images is important for human interpretation. In many cases, the images have lack of clarity and have some effects on images due to fog, low light and other daylight effects exist. So, the images which have these scenarios should be enhanced and made clear to recognize the objects clearly. Histogram-based image enhancement technique is mainly based on equalizing the histogram of the image and increasing the dynamic range corresponding to the image. The Histogram equalization algorithm was performed and tested using different images facing the low light, fog images and color contrast and succeeded in obtaining enhanced images. This technique is implemented by averaging the histogram values as the probability density function. Initially, we have worked with the MATLAB code on Histogram Equalization and made changes to implement in Application Program Interface i.e., API using Xamarin software. The mobile application developed using Xamarin software works efficiently and has less execution time when compared to the application developed in Android Studio. Debugging of the application is successfully done in both Android and IOS versions.

The focus of this thesis is to develop a mobile application on Image enhancement using Xamarin on low light, foggy images.

KEYWORDS: Android, Application Program Interface, Histogram Equalization, Image enhancement, IOS, MATLAB, Probability density function, Xamarin Software.

ACKNOWLEDGEMENT

Firstly, we would like to express our sincere gratitude to our thesis supervisor Dr. Benny Lövström for the continuous support in our thesis work and research. His guidance helped us in our research and learning new Software technology on making a mobile application. We could not have imagined a better support from our supervisor for the thesis work.

We would like to thank our parents and friends for supporting us in every aspect of our education and giving us strengths and courage to overcome difficulties throughout our life.

CONTENTS

List of Figures

1. INTRODUCTION

1.1. Introduction

Here we are dealing with enhancement of an image. Image enhancement plays a very vital role in image processing. Image enhancement leads to manipulation of an image in a suitable way i.e. we obtain more preferable images than the original image. Not all algorithms used for image enhancement make the image better. Image enhancement algorithm should be selected according to the image depending on several factors. So, algorithm has chosen to be in such a way that it gives a better image.

We usually have a problem with low-intensity images and images with less clarity.

The aim of the project is to increase the contrast and color to improve the quality of image. The technique we used in image processing is analyzing the histogram of an image. The algorithms cannot directly have applied to image because most of them lead to several vast modifications such as light condition changes and color temperature alterations.

Most of the image enhancement algorithms work with the non-compressed image or perceptually lossless compression version. In this report, we are presenting the method to enhance the selected image using Histogram Equalization. Using the Histogram Equalization method, we are trying to increase the contrast level and enhance the grey scale or color images.

1.2. Objectives

The main goal of this research is to implement the Image enhancement technique i.e., Histogram Equalization technique as an application in Mobile phones. The low light image or the dark image must be modified and obtained image must be better than the original version. The application should be developed in both Android and IOS in Xamarin environment using C#.

- Histogram Equalization in MATLAB
- Application on Histogram Equalization in Android using Xamarin

- Application on Histogram Equalization in IOS using Xamarin

1.3. Division of work

Algorithm was primarily implemented and verified in MATLAB. Later it was developed into mobile application in Xamarin environment. So, research was developed and implemented in MATLAB by Surya Deepthi and later the application from the algorithm was developed by G.V. Sarath in Xamarin and For developing a mobile application algorithm has to be written in C#. So Deepthi helped in developing C# Code.

1.4. Research Questions

- How well can an algorithm be designed for enhancement of an image using histogram equalization?
- How to design a mobile application efficiently to enhance an image?

1.5. Hypothesis

- ✓ Histogram Equalization is efficient to enhancement of an image because this method usually increases the contrast of an image especially when the image pixels hold very low contrast values.
- ✓ The mobile application is developed in Xamarin environment using C#. The efficiency of the Android and IOS applications are measured in real time. Performance of the application is faster than MATLAB.

1.6. Implementation process

At the beginning, we implemented the histogram equalization method in MATLAB and later, the same algorithm was implemented it in Xamarin visual studio using C# to create mobile applications in both Android and IOS environment.

1.7. Main contribution

After observing all the related surveys, Histogram Equalization algorithm is better for Image enhancement method and it is applicable for different types of RGB (Red, Green, Blue) images. There will be no loss in intensities while converting the RGB image into HSV (Hue, Saturation, Value) image. This technique is applicable for low light images. We get the enhanced image both in MATLAB and in the application in Xamarin environment.

1.8. Outline

The structure of our thesis report is classified into 7 Chapters including current chapter1 of introduction.

- The second chapter deals with image processing.
- The third chapter deals with the main part of this thesis about Image enhancement.
- The fourth chapter includes the modeling of problem solution for implementing in both MATLAB and mobile application in both Android and IOS environments.
- The fifth chapter deals with results and discussion of both MATLAB and mobile application.
- The sixth chapter deals with the conclusion.
- The seventh chapter includes the future scope.
- The eighth chapter includes the references.

2. Image Processing

2.1. Image

An image is a form of representation of a person or anything. An image is a picture that is created or stored electronically. An image can be determined in terms of vector graphics or raster graphics.

- Raster graphics or bitmap image is a representation of an image in a rectangular grid of pixels. It is basically a dot matrix data structure. Raster images are stored in varying formats.
  A bitmap is a single-bit raster, corresponding a bit for bit with image being displayed on the screen. In computer graphics, when the domain is rectangle a bitmap gives a way to store a binary image, that is an image in which each pixel is only two colors.
- The Vector image is a representation of an image in computer graphics by use of polygons. Vector graphics are based on vectors, these vectors lead to locations which are called nodes or control points. These points are positioned with x, y, and axis and determine the direction of the path and may represent different attributes of the image.

Images are stored electronically in many forms such as JPEG, PNG or TIFF.

- JPEG is a graphic image file. It is the format for compressing image i.e. it gives a 'lossy compression' for digital images. The degree of compression can be adjusted.
- PNG (Portable network graph) is basically a raster graphics file that supports lossless data compression. This lossless image compression format is widespread these days on the internet.
- TIFF (Tag Image File Format) is often used a form for exchanging bitmap between applications.TIFF files are commonly used in desktop publishing, faxing, 3-D applications, and medical imaging applications. TIFF files can be of many kinds.

2.2. Image processing

Image processing is analysis and manipulation of an image; this includes improvement of an image, enhancement of the image, compression of image and much more. Image processing is processing of images using mathematical operations using signal processing for which the input is an image and output after image processing may be either a manipulated image or a set of characteristics or parameters related to the image. Images can be processed as three-dimensional signals with the third dimension being time or the z-axis. Image processing is done in three forms digital image processing, optical image processing and analog image processing.[1]

- Analog image processing refers to manipulation an image by processing the signal on two-dimensional analog signals. The analog image represented by analog wave formats e.g.: television broadcasting or live streaming.

- Optical image processing is the field of optical engineering which refers to focusing i.e. focuses on applications of optics. Optical image processing mainly refers to the processing of the image with properties of light. Optical image processing is used in optical instruments such as different types of lenses (concave or convex) or lenses, microscope, telescopes, and other equipment's that use properties of light.

- Digital image processing is the use of algorithms for processing an image (digital images). Digital image processing involves filtering, image padding, and image coding etc. digital image processing. Digital image processing belongs to digital signal processing and is way advantageous then analog signal processing as it has a huge range of algorithms and avoids problems such as image distortion or buildup of noise. Digital image processing consists of

    - Digital imaging
    - Image analysis
    - Image compression
    - Image sharpening
    - Image smoothing
    - Multidimensional systems and much more

3. IMAGE ENHANCEMENT

3.1. Image Enhancement

The main aim of the image enhancement is to improve the interpretability or perception of information in images for human viewers or to provide better input for other automated image processing techniques. Enhancement of the image takes place, in contrast, edges based on the required output. Contrast is an important factor in any subjective evaluation of image quality. In visual perception, contrast is determined by the difference in the color and brightness of the objects and the background of the object with other objects. The Enhancement will not increase the original information, but it will increase the dynamic range of specific features.

Image Enhancement techniques can be divided into two broad categories.

1. Spatial domain methods: It refers to the average of pixels composed of the image and operates directly on these pixels.

   $$g\,(x, y) = T\,[f\,(x, y)].$$

   Where $g\,(x, y)$ is the input of image data

   $f\,(x, y)$ is the processed image data and

   T is the operation used on a set of input images.

2. Frequency domain methods: It operates on the Fourier transform of an image. so, the processed image is obtained by the convolution of an image $f\,(x, y)$ with $h\,(x, y)$.

   $$g\,(x, y) = f\,(x, y) * h\,(x, y)$$

   By applying convolution theorem, the frequency domain is obtained.

   So, the frequency domain with DFT is performed in 2-Dimensional Convolution.

   $$G\,(u, v) = F\,(u, v)\,H\,(u, v)$$

3.2. Color Images

The appearance of an object has basically resulted from the nature of the light reflected from the object, its optical characteristics, and the human perception. The colors are electromagnetic waves described by their wavelength. There are four main attributes that characterize the light: intensity, radiance, luminance, and brightness. In the case of achromatic light, the intensity is the only attribute involved. This is the case where the called gray scale is used: intensity varies from black to white (gray levels in between). The attribute of a visual sensation according to which an area appears to emit more intense light.

In color image analysis three attributes are used to differentiate one color from another: brightness, hue, and saturation. The hue attribute brings the information concerning the main wavelength in the color. The saturation describes the level of a mixture between the hue and the white light. Low values of saturation result in more gray-scale pixels and high values result in pixels with “high color purity”. Finally, the chromaticity is a description that combines hue and saturation. Hence, it is possible to describe an image according to brightness and chromaticity.

3.2.1. RGB Color Model

The RGB (Red, Green, and Blue) color space is one of the most used color spaces, especially for 8bit digital images. This model is usually used for representing colors in electronic devices as TV and computer monitors, scanners, and digital cameras. The RGB is an additive model where the red, green, and blue colors are combined with different quantities or portions to reproduce other colors. The pixels of an image represented in the RGB model have usually 8 bits depth, resulting in 256 possible intensities, i.e., the range of [0, 255] for each color.

A color in the RGB model can be described indicating the amount of red, green, and blue. Each color can vary between the minimum value (totally dark) and the maximum value (totally intense). When all the colors have the minimum value, the resulting color is black. On the contrary, when all the colors have the maximum value, the resulting color is white.

This model is known as the RGB color cube because the model is based on the Cartesian coordinate system and its color subspace of interest is a cube. The primary and secondary colors are at the corners of the cube. The black color is at the origin and the white color is at its opposite corner. The diagonal between the black and the white colors is the grayscale.

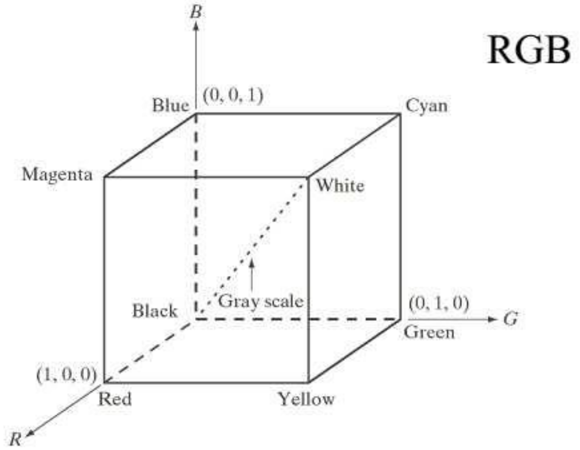


Figure 1. RGB Color model

3.2.2. HSV Color Model

The HSV color system has three components: hue, saturation, and value. This model is also known as HSB (hue, saturation, and brightness). These three parameters are used to define the color space as explained before. The possible values for the hue attribute range from 0 to 360 and the values for the other two attributes range from 0 to 100.

The HSV model is based on cylindrical coordinates and it is a nonlinear transformation of the RGB system. Hence, it is possible to transform directly a color from the HSV system to the RGB system, and contrariwise. There are two other color systems related to HSV: the HSL (Luminosity) system and the HIS (intensity) system. [2]

This color system is very interesting because it allows the separation of the three components of a specific color (hue, saturation, and intensity). It is broadly used in artificial vision systems, as it is a powerful tool for the development of digital image processing algorithms based on the human color perception model. Indeed, the HSV model is well suited to characterize colors in practical terms for human interpretation, differently from the RGB and CMYK models.

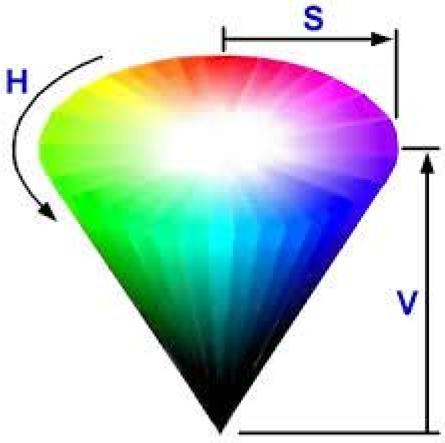


Figure 2. HSV cone model

3.3. Image Formats

There are two main classes of visual information that can be stored in a digital computer the vector and the raster images. The vector format is ideal to present the information that can be described by the junction of simple geometric functions: segments, circumferences, text, and color. In the case of the raster images, the basic element is a value associated with a position i.e. color recorded to a pixel.

In the following, the two image formats are presented in more detail.

1. Vector Formats

The core R graphics engine was essentially vector based. The plots were produced very accurately on vector-based devices. Vector formats have the advantage of being superior when the images need to be visualized at many different scales, however, the files produced are usually of greater size, mainly in the case of very complex images. When using vector formats, the original data resolution and form can be maintained, and the graphics output is usually more refined and elegant.

2. Raster Formats

In the context of digital images, an image format is a common manner to organize and store image data. The format defines how the data are arranged and the used compression type or level. Raster graphics or bitmap files (map of bits) contain a representation of a graphic stored as pixels at a fixed resolution. A common example is a digital photo or a scanned image. Some of the most usual raster formats are GIF, JPEG, PNG, TIFF, and BMP.

a. TIFF

TIFF (*Tagged Image File Format*) is a flexible format that usually stores 8 bits or 16 bits per color (red, green, blue) for a total of 24 or 48 bits, respectively. The extensions used are TIFF or TIF. This format supports several image compression patterns, including JPEG, JPEG-LS and JPEG-2000. There are different common image readers, but not all of them can read all kinds of TIFF files. Consequently, this format is mostly regarded as a family of formats than as a unique one. The data inside TIFF files can be *lossless compressed* or *lossy compressed*.

b. JPEG

JPEG (*Joint Photographic Experts Group*) files to store data in a format with a loss (in major cases). Almost all digital cameras can save images in JPEG format, which supports 8 bits per color for a total of 24 bits, usually producing small files. When the used compression is not so high, the quality of the image is not so much affected, however, JPEG files can suffer from noticeable degradations when edited and saved recurrently. For digital photos that need repetitive edition or when small artifacts are unacceptable, formats without loss besides JPEG should be used for a better storage. This format is also used as the compression algorithm for many PDF files.

c. PNG

The PNG (*Portable Network Graphics*) format was created as a free and open source version of GIF. This format supports true color (16 million of colors) while GIF only supports 256 colors. PNG stands out when an image is formed by large uniformly colored areas. The lossless PNG format is more appropriate for the edition of figures and the lossy formats, as JPEG, are better for final distribution of photos, because JPEG files are smaller than PNG files. Many old Web browsers do not support PNG format; however, all new browsers support most common variations of this format, including transparency in 8 bits.

d. BMP

BMP (*Windows Bitmap*) supports graphics files inside the Microsoft Windows Operational System. Typically, BMP files data are not compressed which results in big size files. The main advantage of this format is its simplicity and broad acceptation.

3.4. Why Histogram Equalization?

Histogram equalization is a method in image processing of contrast adjustment using the image's histogram. This method usually increases the local contrast of many images, especially when the usable data of the image is represented by contrast values. Basically, this method is especially useful in images with background and foreground that are both bright or both dark at the same time. A key advantage of this method is that it is straightforward technique and an invertible operator.[3] If the histogram equalization is known, then the original histogram can be recovered. Histogram equalization is an image-processing technique often used to achieve better quality images in black and white color scales in medical applications such as digital X-rays, MRIs, and CT scans. It is one of the most commonly used algorithms to perform contrast enhancement due to its simplicity and effectiveness. In scientific imaging, where spatial correlation is more important than the intensity of a signal, in this case, histogram equalization works efficiently when compared to other methods. To design an application, the main parameter we consider is that it must take less time to process. So, Histogram Equalization is preferred.[4]

3.5. Histogram Equalization

The method usually increases the contrast of many images, especially when the usable data of an image is represented by close contrast values. Through this adjustment, the intensities can be better distributed on the histogram. The histogram of an image is the representation of the number of pixels that have every value of color. In a greyscale image, it is usually represented as a graphic of the grey values, and in RGB images, the representation is one with three graphics, one for every color component i.e., Red, Green, Blue.

Here the main aim of the histogram equalization is to relate an input image with output image in the histogram. So, to avoid dependence between the number of pixels or the number of quantization levels and the size of the histogram, usually, the histogram axis are normalized between 0 and 1.

Histogram Equalization is a method created to get the advantage of the whole dynamic range in an image. This technique consists in evaluating the probability of every level and then reassigns a new level based on this probability. So, the result would be a flat histogram, where all levels have the same probability, but practical results indicate that this is not true, because of the discrete nature of the data. The new image will have more contrast than the original picture.

In histogram equalization technique, probability density function (pdf) that is being operated, to make it simple changes the pdf of a given image into uniform pdf that spreads out the lower pixel (0) to highest pixel value (n-1) in this way the image is evenly throughout the dynamic range in between 0 and 1. This will be achieved when the pdf is in continuous function. But for the digital image pdf will be a discrete function.[5]

The probability density function based on the histogram is defined as total number of pixels with a specific intensity to the total number of pixels in an image.

$$\text{i.e., pdf (x)} = p(d_k) = \frac{total\ pixels\ with\ intensity\ \text{dk}}{total\ pixels\ in\ image\ X}$$

The cumulative density function is obtained as

$$\text{cdf (x)} = \sum_{k=0}^{n-1} p(d_k)$$

$$\text{So, p } (s_k) = \sum_{k=0}^{n-1} p(d_k)$$

The output of the pixel from the histogram equalization is then equaled to the CDF of the image.

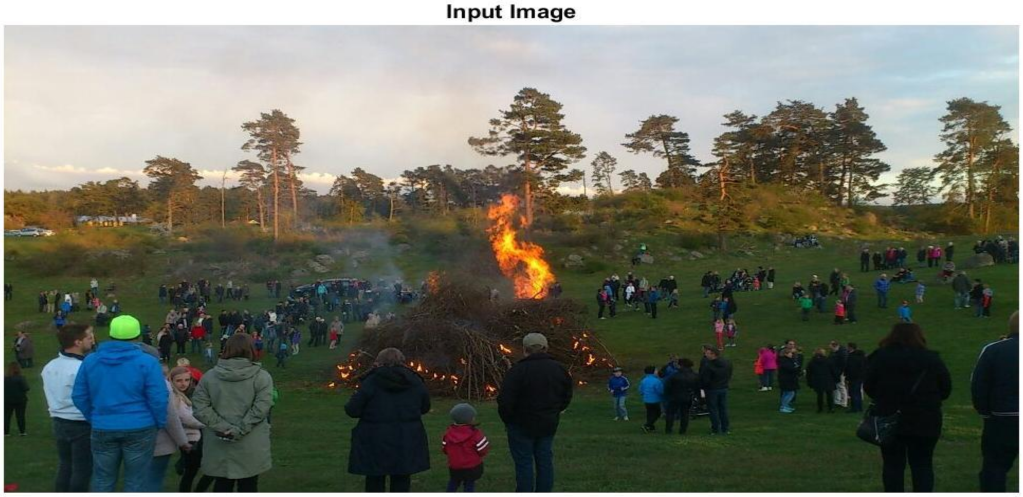


Figure 3. Daylight Input Image

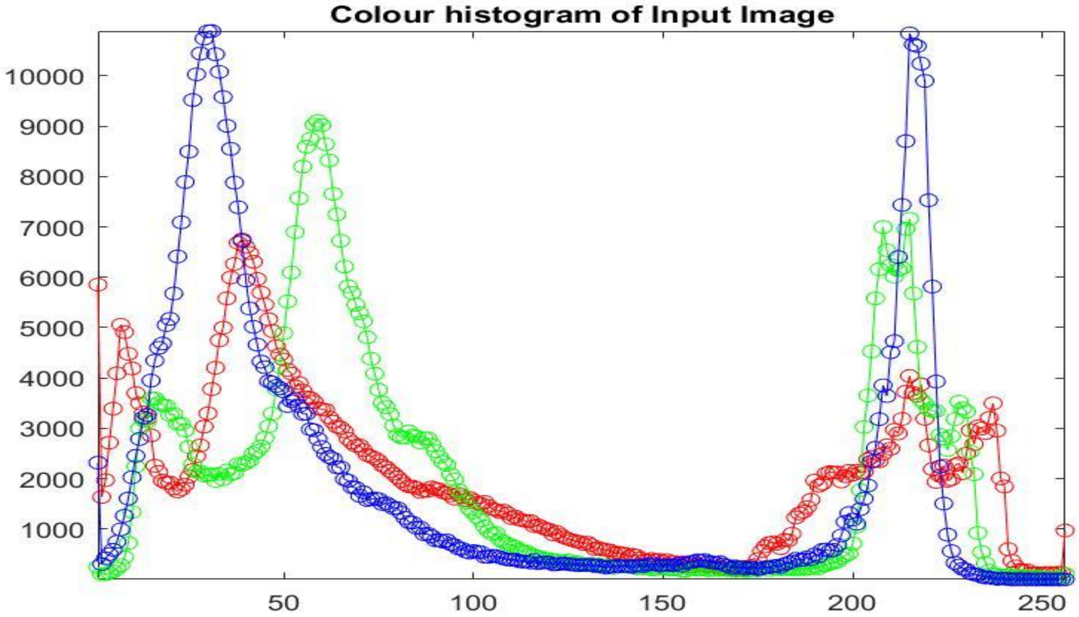


Figure 4. Color Histogram of Daylight Input Image

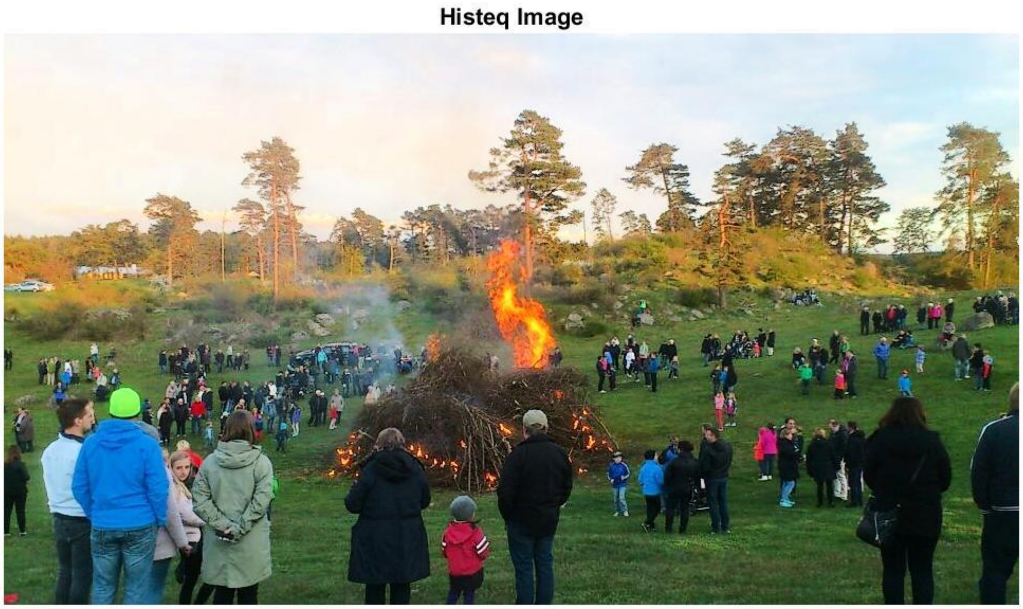


Figure 5. Modified Histogram Equalization Image

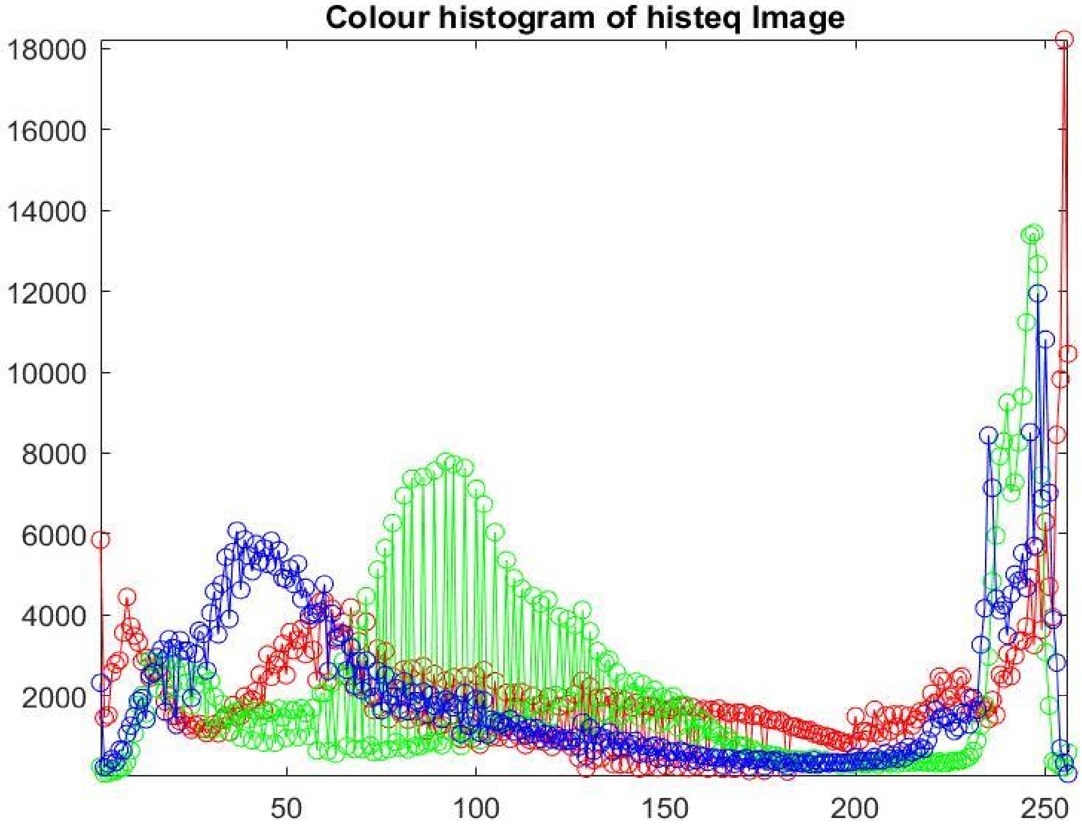


Figure 6. Color Histogram of Modified Image

Histogram Equalization using V component from HSV:

The image must be converted to HSV color space i.e., Hue, Saturation, Value.

- H (hue): The attribute of a visual sensation according to which an area appears to be like one of the perceived colors red, yellow, green, blue or to a combination of two of them.

- S (saturation): The colorfulness of an area judged in proportion to its brightness.

- V (value): The attribute of a visual sensation according to which an area appears to emit intensity of light.

In this method, Histogram Equalization is applied on V component of HSV color space. We must put hue and saturation values as constant and only V must be changed to get histogram equalization image.[6]

H → H
S → S
V → V(equalize)
So, HSV(equalize) → H, S, V(equalize)

4. IMPLEMENTATION

4.1. MATLAB:

MATLAB is a high-performance language for computing. Millions of engineers and scientists worldwide use MATLAB to analyze and design the systems and products transforming our world. The matrix-based MATLAB language is the world's most natural way to express computational mathematics.

MATLAB helps you to take your ideas beyond the desktop. MATLAB can be integrated with other languages, enabling you to deploy algorithms and applications within web, enterprise and production systems. The tools and capabilities of MATLAB are rigorously tested and designed to work together.

The available toolboxes are signal processing, neural networks, fuzzy logic, control systems and many other. MATLAB includes mainly computation, Algorithm development, simulating, engineering graphics and visualization.

4.1.1.1. Histogram Equalization implementation in MATLAB

The proposed algorithm has been designed by considering the RGB image and converting that RGB image into HSV image. The HSV image is divided into different planes which are Hue, Saturation and Value planes. So, then applying the algorithm 'V' plane where the brightness value exists in HSV image. Next reshaped HSV image is converted into RGB image. The modified image is obtained. [7]

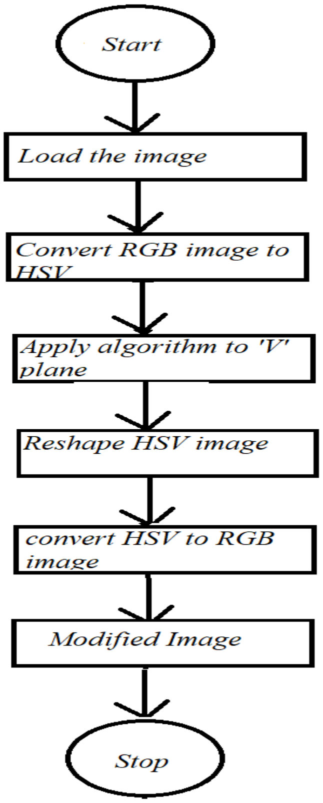


Figure 7. Block diagram of Implementation

Explanation of the flowchart:

1. Firstly, we must load the input image and read it by using ‘imread’ command.
2. Convert RGB image to HSV image by using ‘rgb2hsv’ command.
3. Histogram Equalization is for contrast Image Enhancement which is applied to the extracted ‘V’ plane.
4. Rearranging the enhanced HSV planes and change new V values by concatenation process.
5. Convert HSV to RGB image by using command ‘hsv2rgb’ and we need to store this RGB image as the final output image.
6. Now show the output Modified image by using ‘imshow’ command.

4.2. Development of a Mobile application

The main motivation of making this Mobile application is to use Xamarin studio. Xamarin studio can produce application for both Android and IOS platforms. While making the Android application only we can parallelly create the IOS application also just by changing library file in IOS form. So, the main advantage of this Xamarin environment compared with other platforms is we can create applications in both Android and IOS.[8]

4.2.1.1. Installation of Xamarin in Visual Studio in Windows:

Xamarin can be installed as a part of a new visual studio installation. To achieve this, we must use the following steps:

1. Download Visual studio community from the Visual Studio page.
2. Double-click the downloaded package to start the installation.
3. Select the Mobile development with .NET workload from the installation screen.
4. While Mobile development with .NET is selected, you need to look the summary panel on the right side. Here, you can deselect the mobile development options that you do not want to install. Usually, by default, all the options are selected.
5. When you are ready to begin visual studio installation, click the install button in the bottom corner. Depending on the edition of the visual studio you are installing, the installation process can take a long time to complete.
6. When visual studio installation has completed, click the Launch button to start the visual studio.[9]

Requirements for installing Visual studio tools for Xamarin:

1. Windows 7 or higher
2. Visual Studio 2015 or 2017 (Community, Professional or Enterprise)
3. Xamarin for Visual Studio

4.2.1.2. Difficulties faced during installation:

We have installed the Visual Studio for Xamarin more than 5 times because we have faced many problems while installing this software. One or another package will be always missing in the visual studio. Many times, we faced the problem with Android SDK package as it is either not installed or unable to access it. So, later we understood that user must have admin rights to access that folder.

Sometimes we were unable to provide internet access to emulator after changing the proxy settings. So, then we can try to increase the target API level and install the Arm system image for it and change the proxy settings of the emulator.

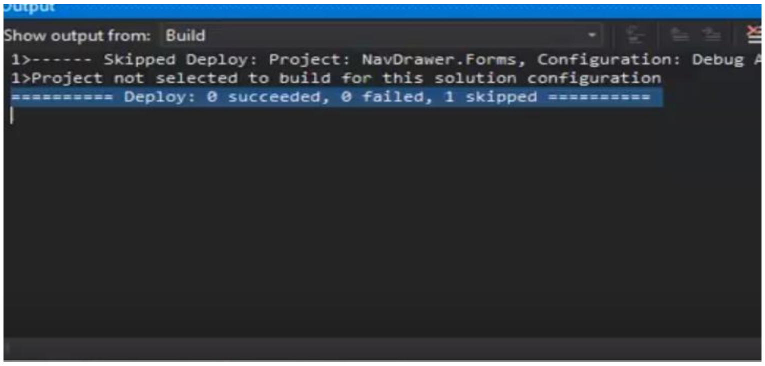
Problem 1: Xamarin installation missing Android SDK

? When we are working on an Android application using Xamarin with visual studio community version. The target platform is Android 7.1, however, after installing the SDK for this platform, the Visual Studio does not show the option to target this version.

➢ We must make sure that the Android SDK location path at Visual Studio’s Tools> options > Xamarin is shown at the top of Android SDK manager.

Problem 2: Deployment errors

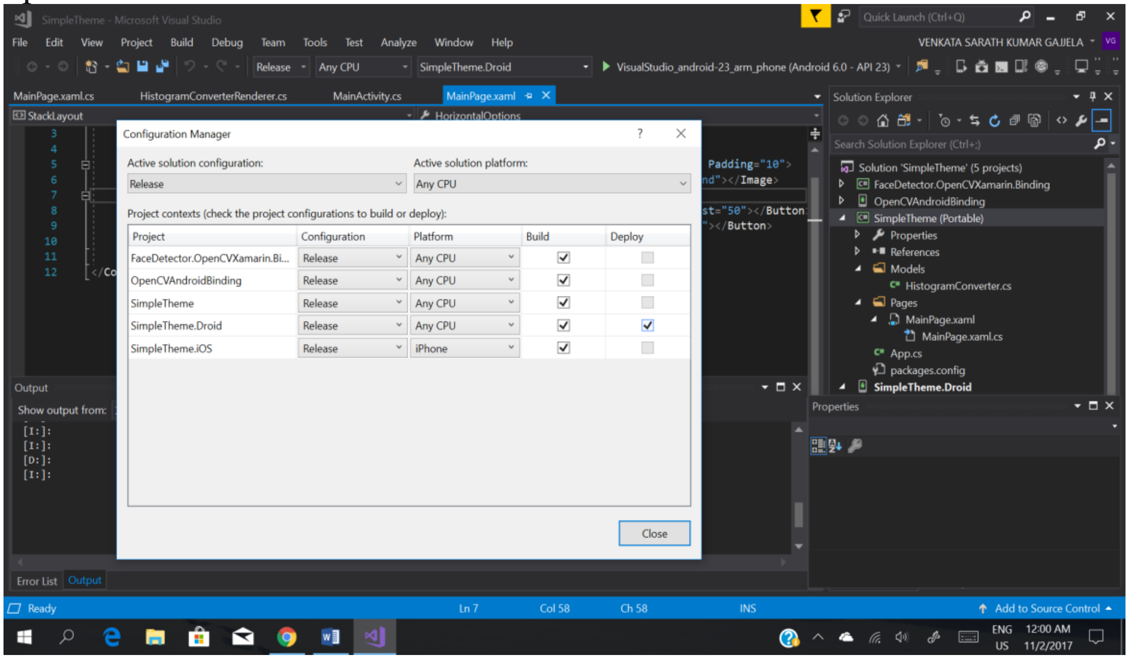
? After building and running the application code, sometimes we get zero errors but showing in the output as 0 succeeded, 0 failed but 1 skipped as below.



```
1>------ Skipped Deploy: Project: NavDrawer.Forms, Configuration: Debug A
1>Project not selected to build for this solution configuration
========== Deploy: 0 succeeded, 0 failed, 1 skipped ==========
```

Figure 8. Output status

- ➢ To solve the above problem, we need to open the configuration manager. So, we must check whether the deploy option is off or not. Mostly the deploying option is off so, we need to turn on the deploy option as below.



Configuration Manager

Active solution configuration: Release

Active solution platform: Any CPU

Project contexts (check the project configurations to build or deploy):

Project	Configuration	Platform	Build	Deploy
FaceDetector.OpenCVXamarin.Bi...	Release	Any CPU	☑	□
OpenCVAndroidBinding	Release	Any CPU	☑	□
SimpleTheme	Release	Any CPU	☑	□
SimpleTheme.Droid	Release	Any CPU	☑	☑
SimpleTheme.iOS	Release	iPhone	☑	□

Figure 9. Configuration Manager

4.3. Histogram Equalization (H.E.) in Windows form

Windows form: The purpose of this topic is to acquaint you with elements of the Visual C# Express Edition integrated development environment (IDE) as you use Windows Forms to build a relatively straightforward C# program. Windows Forms provide your project with components, such as dialog boxes, menus, buttons, and many other controls, that make up a standard Windows application user interface (UI). Fundamentally, these controls are just classes from the .NET Framework class library. The designer view in Visual C# Express Edition enables you to drag the controls onto your application's main form and adjust their size and position. As you do this, the IDE automatically adds the source code to create an instance of the appropriate class and initialize it.

You can create your own web browser application, which you can customize with shortcuts to your favorite websites. You can complete the tasks as below:

- ➢ Create a new Windows forms application.
- ➢ Toggle between code view and designer view.

- Change the Windows form’s properties.
- Add a Menu Strip control.
- Add a Button control.
- Create event handlers for controls.

The algorithm for histogram equalization in windows form is designed and implemented as follows.

- “Select image” button
- “HistEq” button
- “Save” button

Implementation:

In this windows form. we must start implementation with the designing as Form1.designer.cs. we created three buttons in the windows form. The buttons are a select image, HistEq and save. These buttons are defined as

- “Select image” button is defined as to select the image from the device and upload in the image form.
- “HistEq” button is defined as the modification of image for which histogram equalization algorithm is applied.
- “Save” button is defined as to save the modified H.E. image in our system.

4.4. Android implementation in Xamarin studio

Xamarin.Android has several Android API level settings that determine your app’s compatibility with multiple versions of Android. It exposes three Android API level project settings:[10]

- Target Framework
- Minimum Android version
- Target Android version

Before you can configure an API level for your project, you must install the SDK platform components for that API level. If your app supports multiple Android versions, your code must include runtime checks to ensure that your app works with the minimum Android version setting.

As the Android platform evolves and new Android versions are released, each Android version is assigned a unique integer identifier, called the API level. Therefore, each Android version corresponds to a single

Android API level. Because users install apps on older as well as the most recent versions of Android, real-world Android apps must be designed to work with multiple API levels.

4.4.1.1. Histogram Equalization Application in Android Mobile

The implementation of Histogram Equalization in Mobile application for Android is designed below in a flowchart. [11]

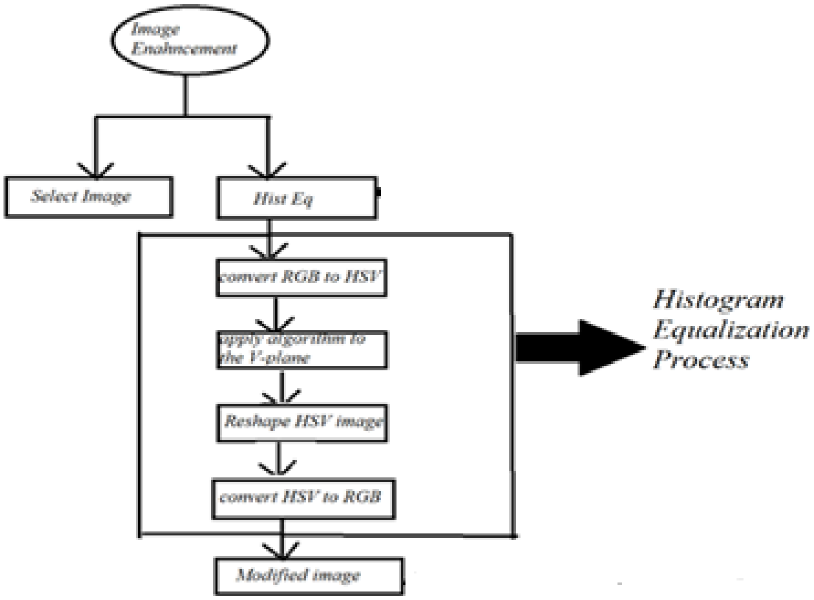


Figure 10. Block diagram for HE application in Android mobile

Explanation of the Flowchart:

1. The Image Enhancement mobile application consists of two buttons namely “SELECT IMAGE” and “HISTEQ” button.
2. If you click the “SELECT IMAGE” button, it loads the image from the device.
3. The other button “HISTEQ” refers to modify the image using Histogram Equalization process.
4. The Modified image is shown in the process is done.

Graphical User Interface:

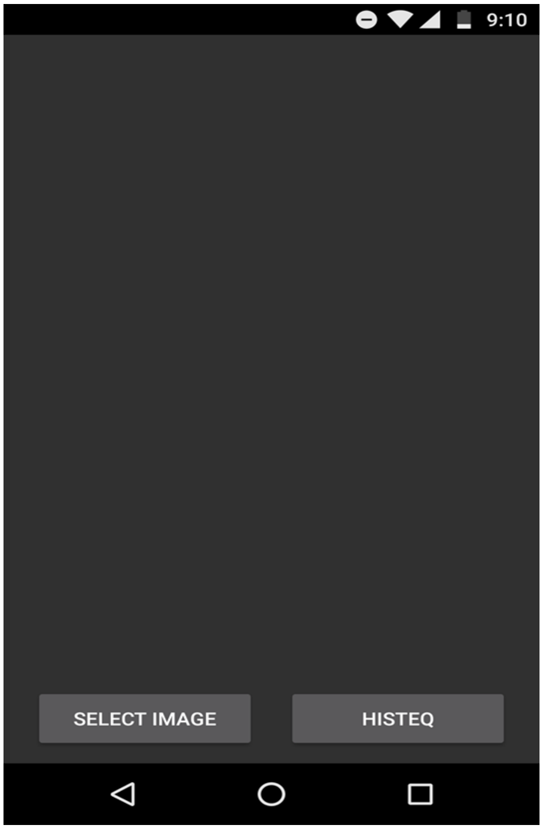


Figure 11. Graphical User Interface of Android application

4.5. IOS implementation in Xamarin Studio

Connecting to the Mac:
To set up the Mac host, you must enable communication between the Xamarin extension for visual studio and your Mac. To do this, allow Remote login on your Mac by the following steps below:

1. Open spotlight and search for Remote Login and then select the Sharing Result. This will open system preferences in the sharing panel.
2. Tick the Remote Login option in the service list on the left to allow Xamarin for the Visual Studio to connect to the Mac.
3. Make sure that Remote Login is set to allow access for All users, or that your Mac username is included in the list of allowed users in the list on the right.[12]

But you must make sure that Xamarin must be installed on your Windows machine.

4.5.1.1. Histogram Equalization Application in IOS Mobile

The implementation of Histogram Equalization in Mobile application for IOS is same as the implementation for Android designed below in a flowchart.

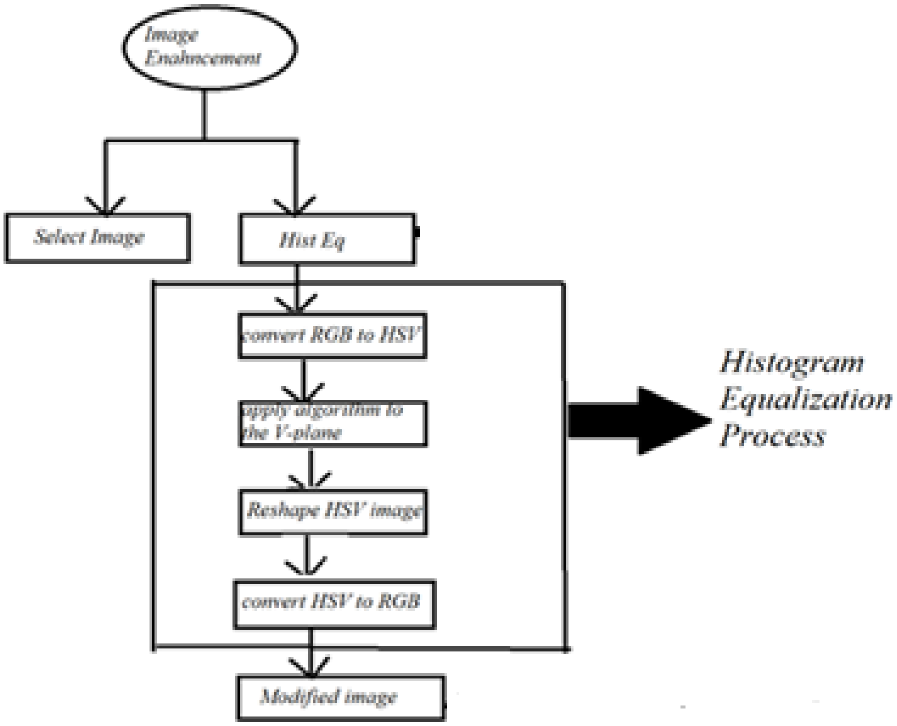


Figure 12. Block diagram of HE application in IOS phone

Graphical User Interface:

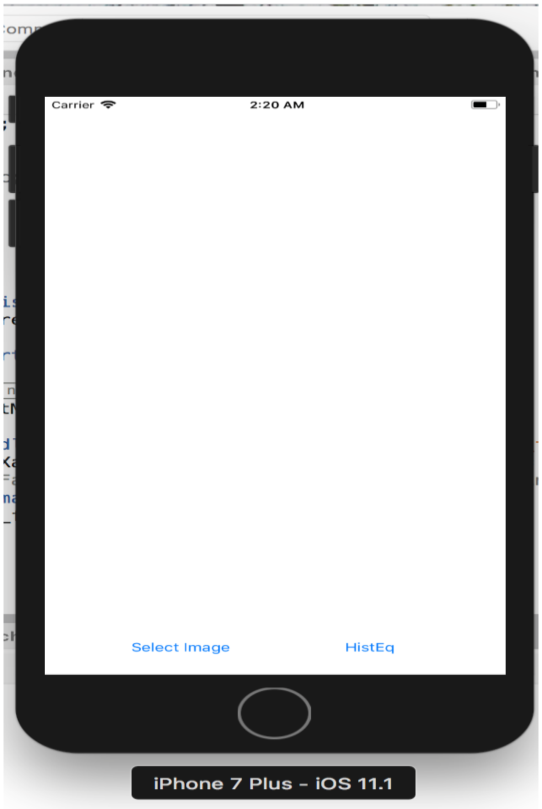


Figure 13. Graphical user Interface of IOS mobile

5. RESULTS AND DISCUSSION

In this results and discussion chapter, we presented the image enhancement through Histogram equalization method. Initially, we show the processed images done in MATLAB. Later we show the processed images which are done in a mobile application using Xamarin environment.

5.1. Using RGB images

The histogram equalization method is one of the best image enhancement technique, these are mainly applied for RGB images. So that firstly we can convert those RGB images into HSV images. Later by using HSV image furtherly, we must put Hue and saturation as constants and Value only must change to maintain the intensities less or more according to the RGB images.

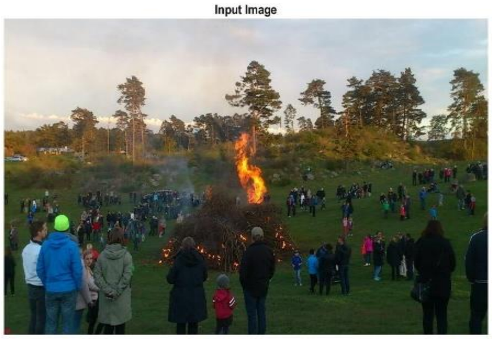


Figure 14. Input RGB image

Later with that HSV image, we must change the V factor in HSV color space to process the image in Histogram Equalization.

5.2. Histogram Equalization

5.2.1.1. Histogram Equalization in MATLAB

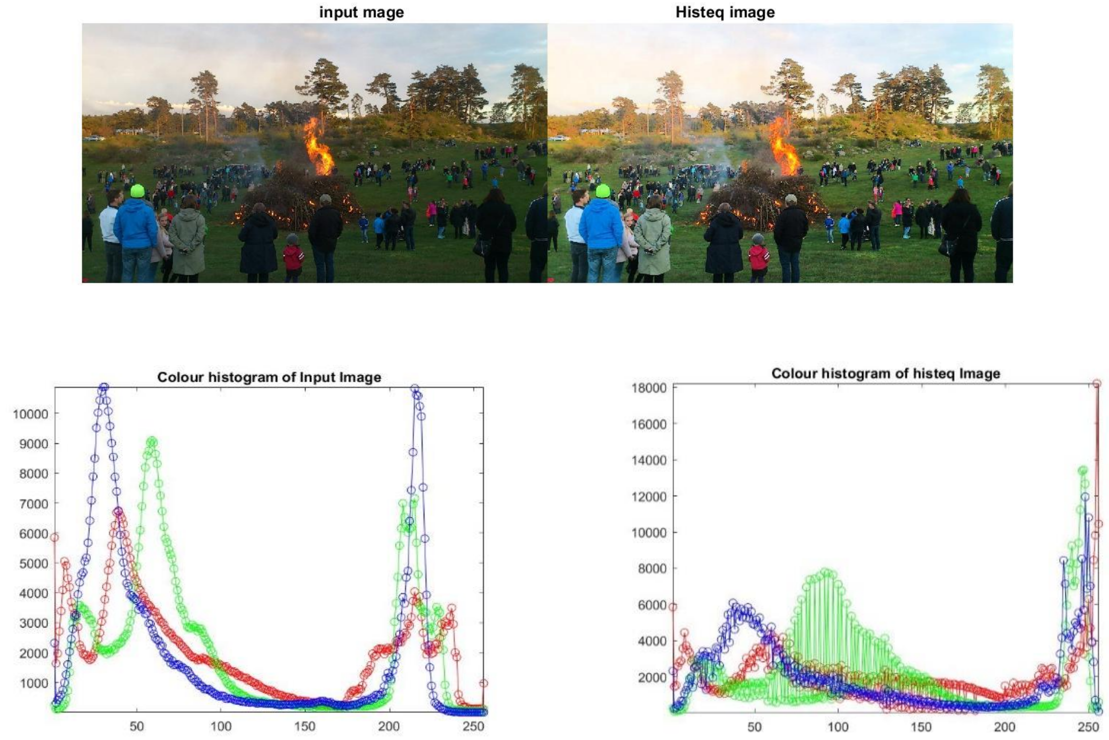


Figure 15. Daylight Input image, Modified images, and its color Histograms

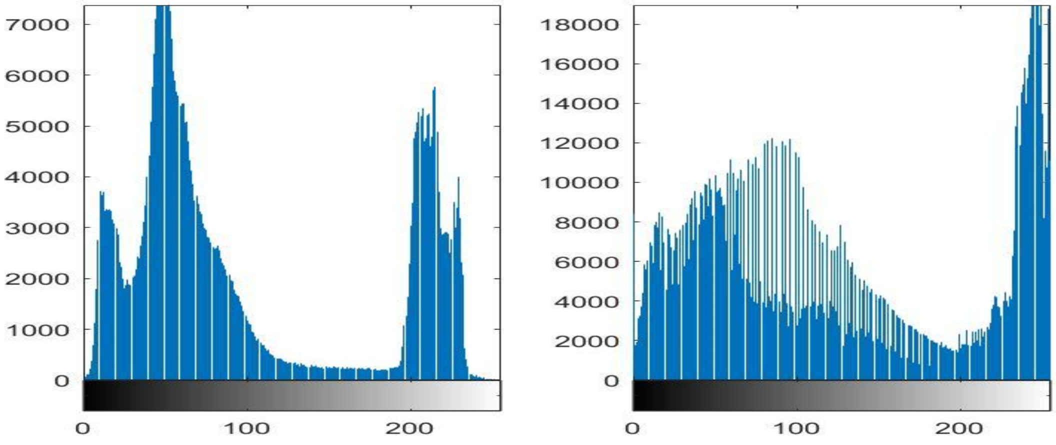


Figure 16. Histograms of Grayscale images of the Original and Modified images.

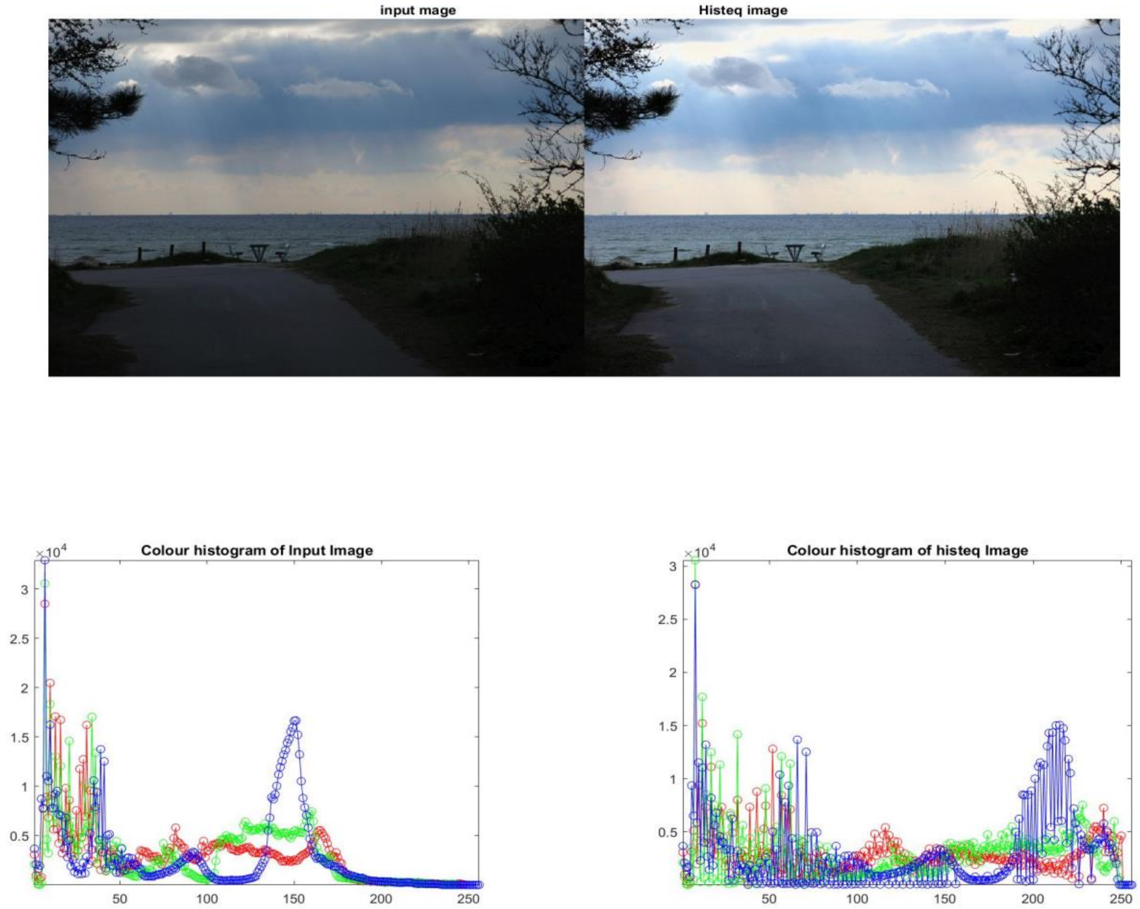


Figure 17. Low light Nature Image, Modified image, and its color Histograms

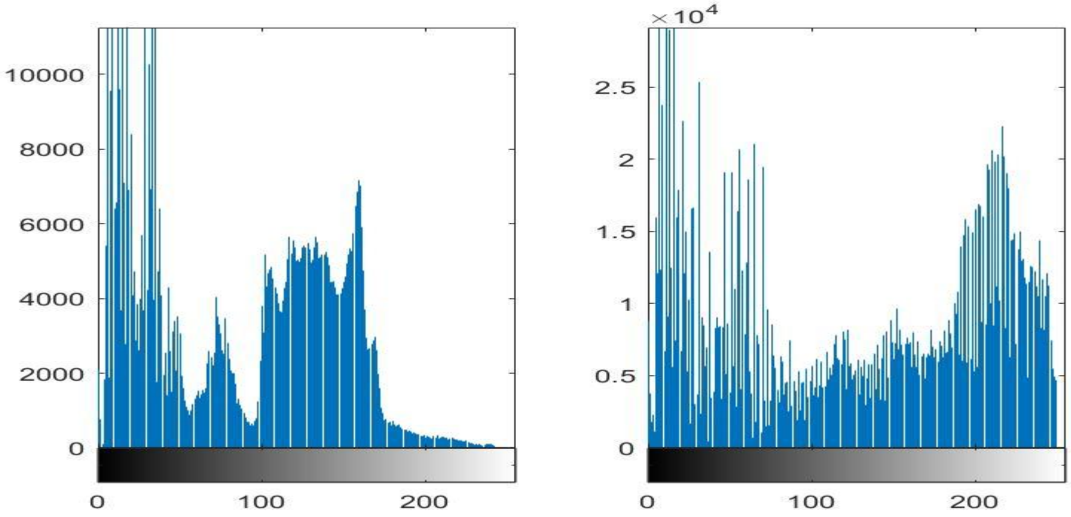


Figure 18. Histograms of Grayscale images of original and Modified images

5.2.1.2. Histogram Equalization in Windows form

After the implementation is done in MATLAB, next we must implement it in a mobile application. So, to make a mobile application we should implement it in Xamarin by using C#. So, the Histogram Equalization method is implemented in C# before doing it in Xamarin. Here we show the results as follows

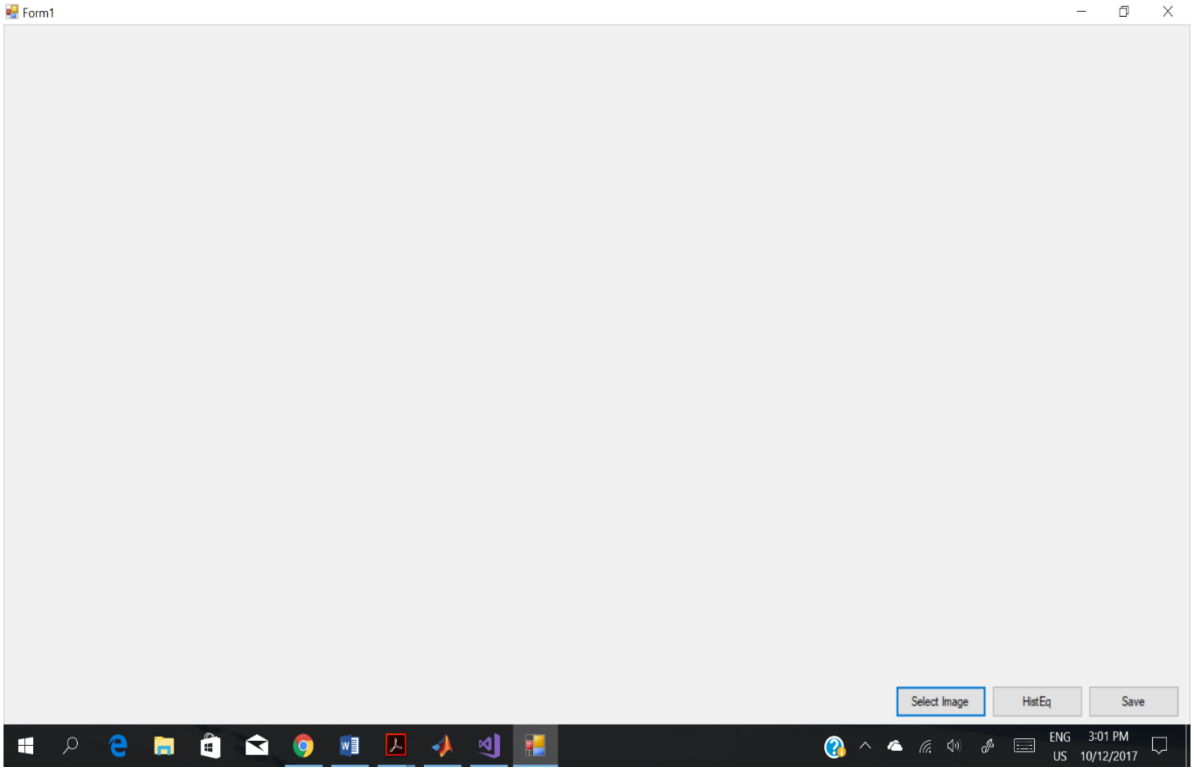


Figure 19. Graphical user Interface of windows form

In the above figure, we can see three buttons Select Image, HistEq and save.

Here Select Image represents to select an image in the system itself.

HistEq represents the modified or processed image using Histogram Equalization method.

Save represents to save the modified image.

The original image is selected by clicking the ‘Select Image’ button.

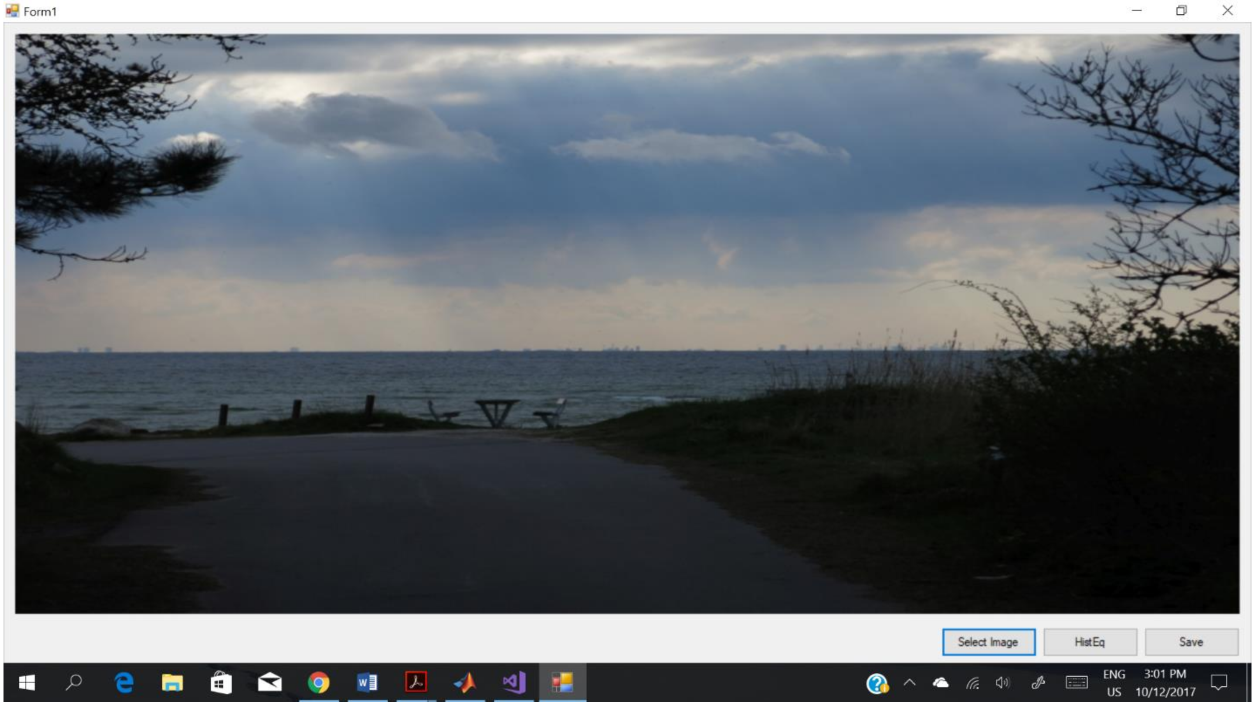


Figure 20. Input Image selection in Windows form

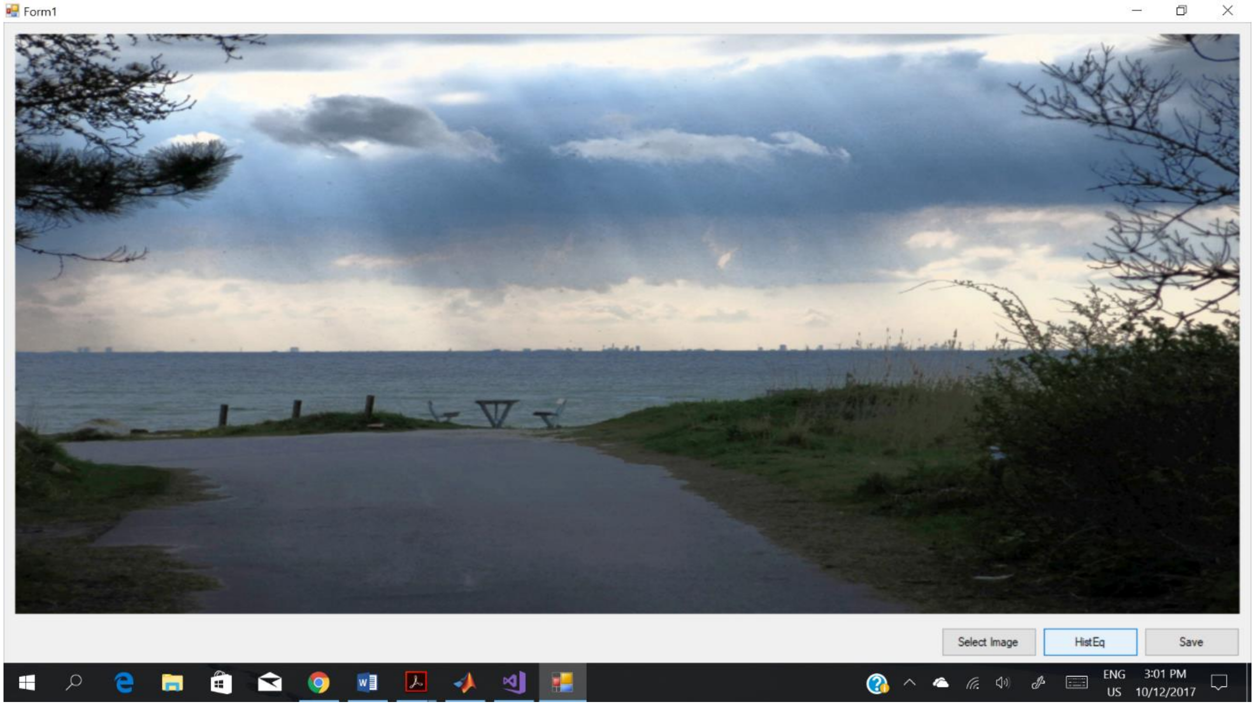


Figure 21. Modified Histeq Image in Windows form

Now by clicking the 'HistEq' button, the modified image is shown. Next, we can see the third button as 'Save'. To save the modified image, we can click the Save button.

5.3. Histogram Equalization Application in Android

5.3.1.1. Emulator

Microsoft Visual Studio 2017 includes an Android studio emulator that can be used as a target for debugging a Xamarin Android app: Visual Studio Emulator for Android. The Visual Studio Emulator for Android can be used as an alternative to the default Android SDK emulator when developing a Xamarin Android application.

After building and debugging the code, firstly Emulator appears like the below figure.

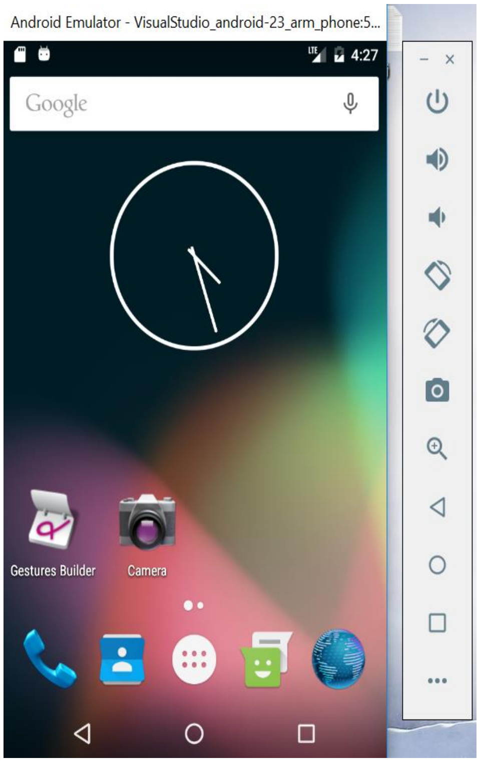


Figure 22. Android Emulator Phone

By opening the application, we can see the application with two buttons Select Image and HistEq.

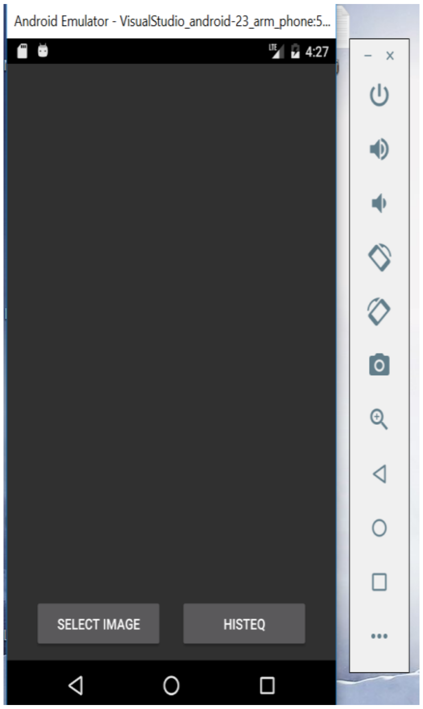


Figure 23. GUI in Android Emulator

By clicking the ‘Select Image’ button the image is selected from the Gallery.

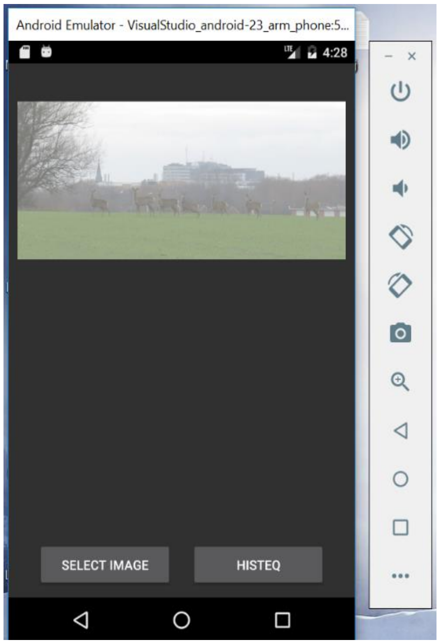


Figure 24. Selection on Input Deer color image

By clicking the ‘HistEq’ button, the modified Histogram Equalization image is displayed at the bottom.

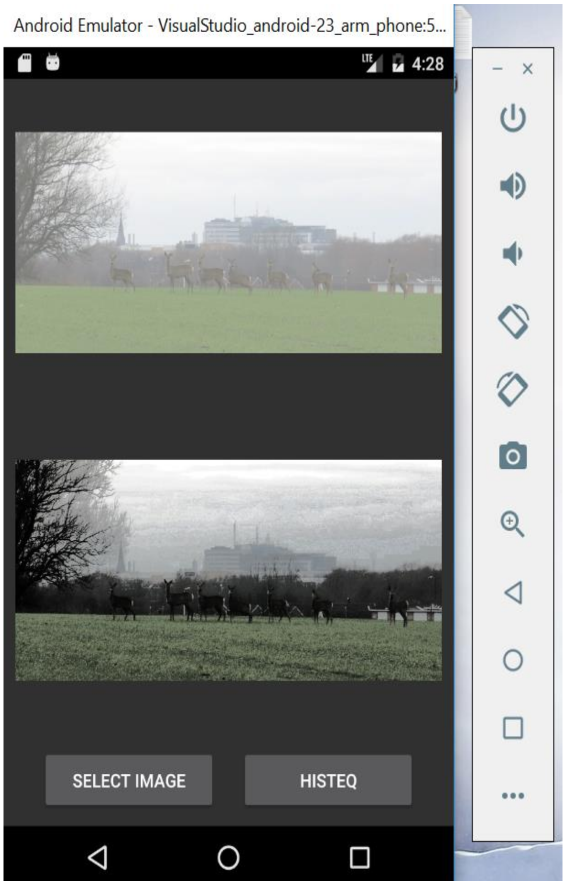


Figure 25. Comparison of both Input and Modified image

5.3.1.2. Android Phone

Loading the input image just by clicking the “SELECT IMAGE” button and we get the modified image by clicking the “HISTEQ” button.

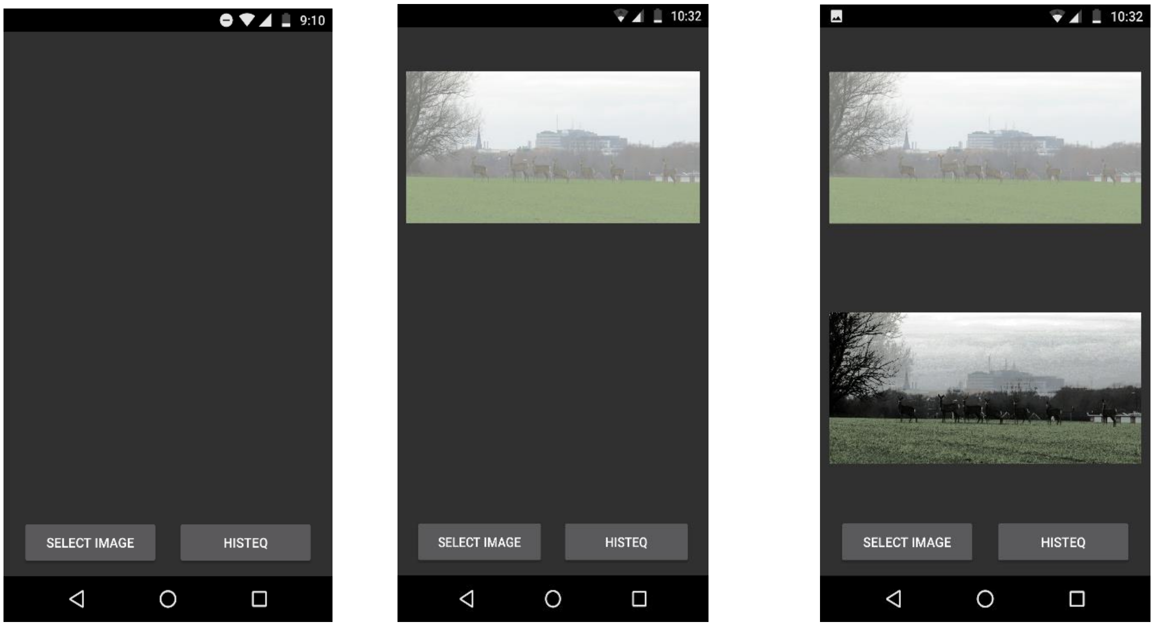


Figure 26. Step by step process of Histeq Application for “Deer image covered with fog” in Android Phone.

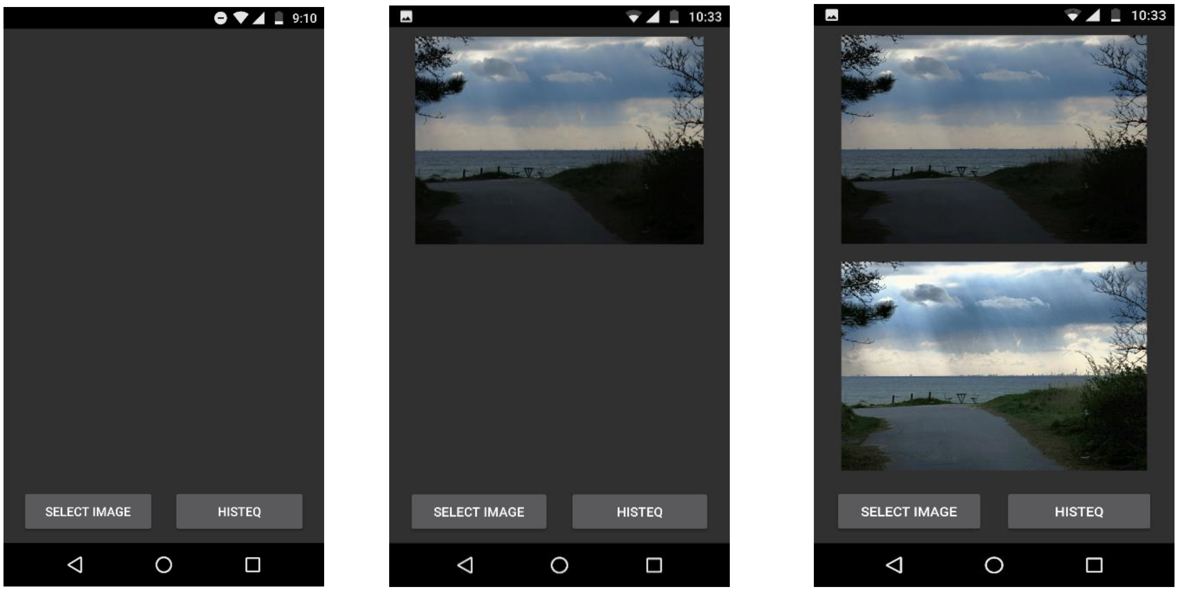


Figure 27. Step by step process of Histeq Application for “Low Light Nature image” in Android Phone.

These are the graphical interfaces of each step of Histogram Equalization Application in Android Mobile. So, the comparison of the selected image and Modified image is shown in the mobile application itself.

5.4. Histogram Equalization Application in IOS:

5.4.1.1. Emulator

Figure 28. IOS Emulator phone

By opening the application, we can see the application with two buttons Select Image and HistEq.

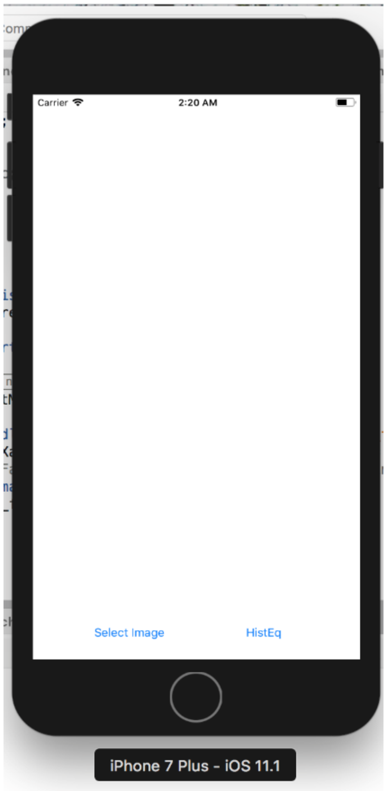


Figure 29. GUI in IOS emulator

By clicking the 'Select Image' button the image is selected from the Gallery.

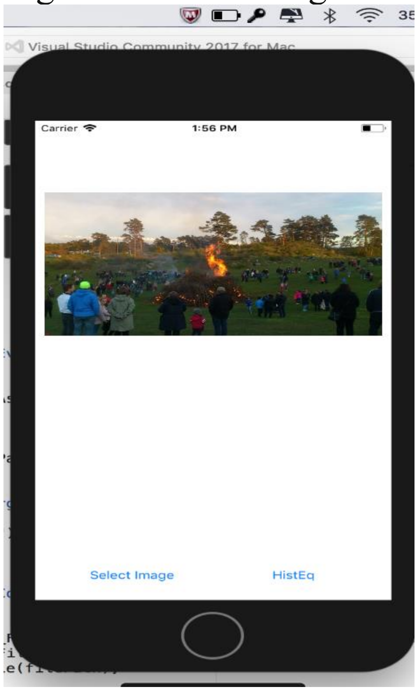


Figure 30. Selection on input image

By clicking the ‘HistEq’ button, the modified Histogram Equalization image is displayed at the bottom.

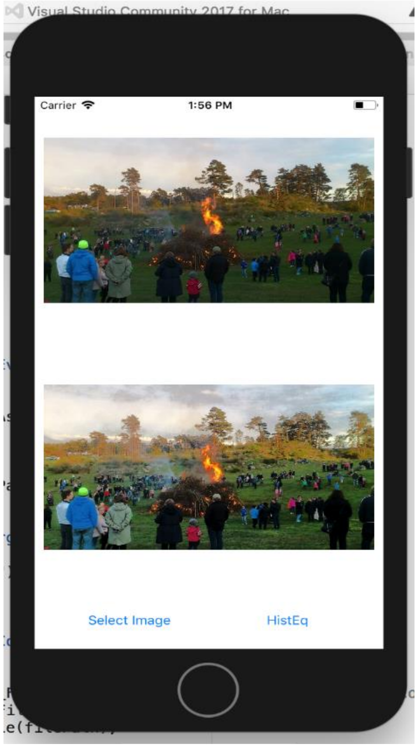


Figure 31. Comparison of both input and Modified image.

6. CONCLUSION

- In this thesis, the image enhancement is applied to both RGB and Grey scale images. The results show the effectiveness of the algorithm in improving both contrast and color of the Original image.
- Histogram equalization is a very good technique as it has the best results with a sufficient increase in contrast and color of the image.
- It also gives the same results while implementing in Android and iOS applications with a low computational power and fast execution.

7. FUTURE WORK

In this research work, comprehensive experiments on image enhancement has conducted on low light, foggy images for only one method in both Android and iOS versions. This work can be further extended to have multiple image enhancement algorithms in the same mobile application itself. So, we can have better compared images than the original image.

8. REFERENCES:

[1] D. S. Sohi and S. S. Devgan, “Application to enhance the teaching and understanding of basic image processing techniques,” in *Proceedings of the IEEE Southeastcon 2000*, 2000, pp. 413–416.

[2] C. Dan, Q. Zhang, H. Wang, and S. Feng, “A novel color image contrast enhancement algorithm using saturation-lightness combined stretching in the HSL color space for AMOLED applications,” in *2014 12th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, 2014, pp. 1–3.

[3] S. Kansal, S. Purwar, and R. K. Tripathi, “Trade-off between mean brightness and contrast in histogram equalization technique for image enhancement,” in *2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, 2017, pp. 195–198.

[4] C. R. Nithyananda, A. C. Ramachandra, and Preethi, “Review on Histogram Equalization based Image Enhancement Techniques,” in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, 2016, pp. 2512–2517.

[5] S. Patel and M. Goswami, “Comparative analysis of Histogram Equalization techniques,” in *2014 International Conference on Contemporary Computing and Informatics (IC3I)*, 2014, pp. 167–168.

[6] P. S. Rajpoot and A. Chouksey, “A Systematic Study of Well Known Histogram Equalization Based Image Contrast Enhancement Methods,” in *2015 International Conference on Computational Intelligence and Communication Networks (CICN)*, 2015, pp. 242–245.

[7] A. Taguchi, “Enhancement of MPEG/JPEG images based on histogram equalization without gamut problem,” in *2008 IEEE Conference on Soft Computing in Industrial Applications*, 2008, pp. 269–272.

[8] M.-C. Lee and S.-B. Cho, “Interactive differential evolution for image enhancement application in smart phone,” in *2012 IEEE Congress on Evolutionary Computation*, 2012, pp. 1–6.

[9] “Installing Xamarin in Visual Studio on Windows - Xamarin.” [Online]. Available: https://developer.xamarin.com/guides/cross-platform/getting_started/installation/windows/. [Accessed: 24-Jan-2018].

[10] “Understanding Android API Levels - Xamarin.” [Online]. Available: https://developer.xamarin.com/guides/android/application_fundamentals/understanding_android_api_levels/. [Accessed: 24-Jan-2018].

[11] J. Lee, J. Ahn, Y. Lee, and C. Kim, “Implementation of Semantic Directory Service on Image Gallery Application of Mobile Devices,” in *2013 International Conference on Information Science and Applications (ICISA)*, 2013, pp. 1–2.

[12] “Connecting to the Mac - Xamarin.” [Online]. Available: https://developer.xamarin.com/guides/ios/getting_started/installation/windows/connecting-to-mac/. [Accessed: 24-Jan-2018].